

# **Generation of bit stream containing binary image/audio data that is multiplexed with code defining object in ASCII format**

**Publication number:** CN1224982 (A)

**Publication date:** 1999-08-04

**Inventor(s):** SUZUKI TERUHIKO [JP]

**Applicant(s):** SONY CORP [JP]

**Classification:**

- international: G06F17/28; G06F19/00; G06T1/00; G06T9/00; G06T15/00;  
G06T17/00; H04L29/00; H04N11/04; G06F17/28; G06F19/00;  
G06T1/00; G06T9/00; G06T15/00; G06T17/00; G11B;  
H04L29/00; H04N11/04; (IPC1-7) H04N11/04

- European:

**Application number:** CN19981024528 19980922

**Priority number(s):** JP19970275196 19970922

**Also published as:**

CN1247029 (C)

EP0903698 (A2)

EP0903698 (A3)

US6611262 (B1)

SG89267 (A1)

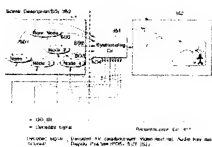
more >>

Abstract not available for CN 1224982 (A)

Abstract of corresponding document: EP 0903698 (A2)

A system controller directs a storage device to output a scene description data corresponding to a user request signal. A parsing circuit extracts a URL (Uniform Resource Locator) included in the scene description data, and causes storage devices to output elementary stream and object stream corresponding to the URL. After extracting an object descriptor from the object stream, a generator generates an ID of this object descriptor and supplies it to an encoder. Further, the generator adds the ID to the object descriptor and outputs the ID-added object descriptor to a multiplexer. The multiplexer multiplexes the scene description data including the ID that has been converted into binary format, the object descriptor and the elementary streams into a multiplexed stream for output thereof.

FIG 10



Data supplied from the esp@cenet database — Worldwide



# 权 利 要 求 书

1. 一种产生由多个节点定义的三维空间模型数据和由各节点中所包含的位置描述的图像/音频数据的方法, 所述方法包括如下步骤:

- 5       从所述三维空间模型数据的节点中取出相应的位置;  
      把取出的位置转换为对应于与所述位置相连系的图像/音频数据的流 ID;

      用所述流 ID 替换所述位置; 以及

- 多路复用所述图像/音频数据和包括所述流 ID 的所述三维空间模型数  
10 据, 以便产生比特流。

2. 根据权利要求 1 的方法, 其中所述比特流包括含有所述流 ID 和定义所述图像/音频数据的附加信息, 所述附加信息已经与所述图像/音频数据和三维空间模型数据在所述比特流中多路复用。

3. 根据权利要求 1 的方法, 其中所述三维空间模型数据采用虚拟现实模  
15 型化语言(Virtual Reality Modeling Language(VRML))描述, 所述位置采用 ASCII 格式的统一资源定位符(Uniform Resource Locator(URL))表示, 所述流 ID 用二进制格式表示。

4. 根据权利要求 3 的方法, 进一步包括把所述流 ID 转换为字符串, 并  
根据所述图像/音频数据是由同一个服务器提供还是由不同的服务器提供确  
20 定是用所述流 ID 还是用所述字符串替换所述图像/音频数据的位置。

5. 根据权利要求 4 的方法, 其中所述比特流包括含有所述流 ID 和定义所述图像/音频数据的附加信息, 所述附加信息已经与所述图像/音频数据和三维空间模型数据在所述比特流中多路复用。

6. 一种产生由多个节点定义的三维空间模型数据和由各节点中的所包  
25 含的位置描述的图像/音频数据的方法, 所述方法包括如下步骤:

      从所述三维空间模型数据的节点中取出相应的位置;

      把取出的位置转换为对应于与所述位置相连系的图像/音频数据的流 ID;

      把所述流 ID 转换为字符串;

- 用所述字符串替换所述位置; 以及  
      多路复用所述图像/音频数据和包括在所述字符串中的所述三维空间模  
30

型数据，以便产生比特流。

7. 根据权利要求 6 的方法，其中根据所述图像/音频数据是由同一个服务器提供还是由不同的服务器提供使所述位置用所述字符串替换。

8. 根据权利要求 6 的方法，其中所述三维空间模型数据采用虚拟现实模型化语言(Virtual Reality Modeling Language(VRML))描述，所述位置采用 ASCII 格式的统一资源定位符(Uniform Resource Locator(URL))表示，所述流 ID 用二进制格式表示。

9. 根据权利要求 6 的方法，其中所述比特流包括含有所述流 ID 和定义所述图像/音频数据的附加信息，所述附加信息已经与所述图像/音频数据和  
10 三维空间模型数据在所述比特流中多路复用。

10. 一种用于产生由多个节点定义的三维空间模型数据和各节点中所包含的位置描述的图像/音频数据的设备，它包括：

用于从所述三维空间模型数据的节点中取出相应的位置的装置；

用于把取出的位置转换为对应于与所述位置相连系的图像/音频数据的

11 流 ID 的装置；

用所述流 ID 替换所述位置的装置；以及

用于多路复用所述图像/音频数据和包括所述流 ID 的所述三维空间模型数据，以便产生比特流的装置。

11. 根据权利要求 10 的设备，其中所述比特流包括含有所述流 ID 和定义所述图像/音频数据的附加信息，所述附加信息已经与所述图像/音频数据和  
20 三维空间模型数据在所述比特流中多路复用。

12. 根据权利要求 10 的装置，其中所述三维空间模型数据采用虚拟现实模型化语言(Virtual Reality Modeling Language(VRML))描述，所述位置采用 ASCII 格式的统一资源定位符(Uniform Resource Locator(URL))表示，所述流  
25 ID 用二进制格式表示。

13. 根据权利要求 12 的装置，进一步包括把所述流 ID 转换为字符串的装置，和如下装置，即根据所述图像/音频数据是由同一个服务器提供还是由不同的服务器提供而确定是用所述流 ID 还是用所述字符串替换所述图像/音频数据的位置的装置。

14. 根据权利要求 13 的装置，其中所述比特流包括含有所述流 ID 和定义所述图像/音频数据的附加信息，所述附加信息已经与所述图像/音频数据

和三维空间模型数据在所述比特流中多路复用。

15. 一种用于产生由多个节点定义的三维空间模型数据和由各节点中所包含的位置描述的图像/音频数据的设备，它包括：

- 用于从所述三维空间模型数据的节点中取出相应的位置的装置；
- 5 用于把取出的位置转换为对应于与所述位置相连系的图像/音频数据的流 ID 的装置；
- 把所述流 ID 替换为字符串的装置；
- 用所述字符串替换所述位置的装置；以及
- 用于多路复用所述图像/音频数据和包括所述字符串的所述三维空间模型数据，以便产生比特流的装置。

16. 根据权利要求 15 的设备，其中根据所述图像/音频数据是由同一个服务器提供还是由不同的服务器提供使所述位置用所述字符串替换。

17. 根据权利要求 15 的装置，其中所述三维空间模型数据采用虚拟现实模型化语言(Virtual Reality Modeling Language(VRML))描述，所述位置采用 ASCII 格式的统一资源定位符(Uniform Resource Locator(URL))表示，所述流 ID 用二进制格式表示。

18. 根据权利要求 15 的装置，其中所述比特流包括含有所述流 ID 和定义所述图像/音频数据的附加信息，所述附加信息已经与所述图像/音频数据和三维空间模型数据在所述比特流中多路复用。

20 19. 一种处理包括图像/音频数据和由多个节点构成的三维空间模型数据的比特流以便产生显示图像的方法，该方法包括：

接收所述比特流；

把接收的比特流多路分解成流 ID、所述三维空间模型数据和图像/音频数据；以及

25 根据所述流 ID 找出所述图像/音频数据与各个节点之间的相应之处，以便产生所述显示图像。

20. 根据权利要求 19 的方法，其中所述三维空间模型数据采用虚拟现实模型化语言(Virtual Reality Modeling Language(VRML))描述，所述流 ID 信息用二进制格式表示。

30 21. 根据权利要求 19 的方法，其中所述图像/音频数据与节点信息之间的相应之处是与由第一表达式表示的所述流 ID 或者与由第二表达式表示所

述流 ID 对应的的字符串一致的；而且其中所述节点包括标记位，该标记位表示根据所述图像/音频数据是由同一个服务器提供还是由不同的服务器提供而使用第一或第二表达式。

22. 根据权利要求 19 的方法，其中所述比特流包括含有所述流 ID 和定义所述图像/音频数据的附加信息，所述附加信息已经与所述图像/音频数据和三维空间模型数据在所述比特流中多路复用，并使其中所述节点中包含所述流 ID 与在所述附加信息中包含的所述流 ID 核对。

23. 一种处理包括图像/音频数据和由多个节点构成的三维空间模型数据的比特流以便产生显示图像的方法，该方法包括：

- 10 接收所述包括图像/音频数据和由节点构成的三维空间模型数据的比特流，并输出所述三维空间模型数据和图像/音频数据；

把字符串信息转换为流 ID 信息，所述字符串信息是表示节点与所述图像/音频数据之间的关系的的信息；以及

根据转换的流 ID 信息连接所述图像/音频数据与所述节点。

- 15 24. 根据权利要求 23 的方法，其中表示所述关系的信息是与所述流 ID 信息对应的所述字符串信息和位置指定信息之一；其中，如果表示所述关系的信息是所述字符串信息，则所述图像/音频数据根据转换的流 ID 信息与所述节点连接；表示所述关系的信息所述位置指定信息的情况下，所述图像/音频数据从由所述位置指定信息指定的提供部分与所述节点连接。

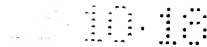
- 20 25. 根据权利要求 23 的方法，其中所述三维空间模型数据采用虚拟现实模型化语言(Virtual Reality Modeling Language(VRML))描述，所述字符串信息采用 ASCII 格式表示，所述转换的流 ID 信息用二进制格式表示。

26. 根据权利要求 23 的方法，其中所述比特流包括定义所述图像/音频数据和包含所述流 ID 的信息；而且其中转换的流 ID 信息与定义所述图像/25 音频数据的信息中所包含的流 ID 核对，所述图像/音频数据根据核对的结果与所述节点连接。

27. 一种处理包括图像/音频数据和由多个节点构成的三维空间模型数据的比特流以便于生成显示图像的设备，包括：

接收所述比特流的装置；

- 30 把接收的比特流多路分解成流 ID、所述三维空间模型数据和图像/音频数据的装置；以及



根据所述流 ID 找出所述图像/音频数据与各个节点之间的相应之处以产生所述显示图像的装置。

28. 根据权利要求 27 的设备, 其中所述三维空间模型数据采用虚拟现实模型化语言(Virtual Reality Modeling Language(VRML))描述, 所述流 ID 用二进制格式表示。

29. 根据权利要求 27 的设备, 其中所述图像/音频数据与节点信息之间的相应之处是与由第一表达式表示的所述流 ID 或者与由第二表达式表示的所述流 ID 对应的字符串; 而且其中所述节点包括标记位, 该标记位表示根据所述图像/音频数据是由同一个服务器提供还是由不同的服务器提供而使用第一或第二表达式。

30. 根据权利要求 27 的设备, 其中所述比特流包括含有所述流 ID 和定义所述图像/音频数据的附加信息, 所述附加信息已经与所述图像/音频数据和三维空间模型数据在所述比特流中多路复用; 并使其中所述节点中包含的所述流 ID 与所述附加信息中包含的所述流 ID 核对。

31. 一种处理包括图像/音频数据和由多个节点构成的三维空间模型数据的比特流以便产生显示图像的设备, 它包括:

接收所述包括图像/音频数据和由节点构成的三维空间模型数据的比特流的装置;

输出所述三维空间模型数据和图像/音频数据的装置;

把字符串信息转换为流 ID 信息的装置, 所述字符串信息是表示节点与所述图像/音频数据之间的关系的; 以及

根据转换的流 ID 信息连接所述图像/音频数据与所述节点的装置。

32. 根据权利要求 31 的设备, 其中表示所述关系的信息是与所述流 ID 信息对应的所述字符串信息和位置指定信息之一; 其中, 如果表示所述关系的信息是所述字符串信息, 则所述图像/音频数据根据转换的流 ID 信息与所述节点连接; 在表示所述关系的信息是所述位置指定信息的情况下, 所述图像/音频数据从由所述位置指定信息指定的提供部分与所述节点连接。

33. 根据权利要求 31 的设备, 其中所述三维空间模型数据采用虚拟现实模型化语言(Virtual Reality Modeling Language(VRML))描述, 所述字符串信息采用 ASCII 格式表示, 所述流 ID 信息用二进制格式表示。

34. 根据权利要求 31 的设备, 其中所述比特流包括定义所述图像/音频

数据和包含所述流 ID 的信息；而且其中转换的流 ID 信息与定义所述图像/音频数据的信息中所包含的流 ID 核对，所述图像/音频数据根据核对的结果与所述节点连接。

35. 一种上面记录有数据生成程序的记录介质，该数据生成程序用于产生由多个节点构成的三维空间模型数据和由所述三维空间模型数据的节点中包含的位置指定信息指定的图像/音频数据，被执行的所述数据生成程序完成如下步骤：

取出所述三维空间模型数据的节点中包含的位置指定信息；

把取出的位置指定信息转换为对应于由所取出的位置指定信息指定的

- 10 所述图像/音频数据的流 ID 信息；

用所述流 ID 信息替换所述节点中包含的位置指定信息；以及

多路复用所述图像/音频数据和所述流 ID 信息中包含的三维空间模型数据，以便产生比特流。

36. 一种根据权利要求 35 的记录介质，其中进一步执行的所述数据生成程序完成如下步骤：

把所述流 ID 信息转换为由第一表达式表示的字符串信息；以及

确定是否用以第二表达式表示的所述流 ID 信息或以第一表达式表示的所述字符串信息替换所述节点中包含的所述位置指定信息，其中根据确定的结果替换所述节点中包含的所述位置指定信息。

- 20 37. 一种上面记录有数据生成程序的记录介质，该数据生成程序用于产生由多个节点构成的三维空间模型数据和由所述三维空间模型数据的节点中包含的位置指定信息指定的图像/音频数据，被执行的所述数据生成程序完成如下步骤：

取出所述三维空间模型数据的节点中包含的位置指定信息；

- 25 把取出的位置指定信息转换为对应于由所取出的位置指定信息指定的图像/音频数据的流 ID 信息；

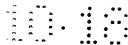
把所述流 ID 信息转换为字符串信息；

用所述字符串信息替换所述节点中包含的所述位置指定信息；以及

- 30 多路传输所述图像/音频数据和包含所述字符串信息的三维空间模型数据，以便产生比特流。

38. 一种上面记录有数据处理程序的记录介质，该数据处理程序用于由





包括图像/音频数据和由多个节点构成的三维空间模型数据的比特流产生显示图像, 执行所述数据处理程序以完成如下步骤:

接收包括所述图像/音频数据和由多个节点构成的所述三维空间模型数据的所述比特流;

5 输出所述三维空间模型数据和图像/音频数据;

根据表示所述节点与所述图像/音频数据之间的关系的的信息, 连接所述图像/音频数据与节点, 所述表示关系的的信息是流 ID 信息。

39. 一种根据权利要求 38 的记录介质, 其中表示所述关系的的信息是由第一表达式表示的所述流 ID 信息和与所述流 ID 信息对应的字符串信息之一;

10 所述字符串信息由第二表达式表示; 而且其中所述节点包括标记位信息, 该标记位标致表示所述关系的之信息的第一或第二表达式, 进一步执行所述数据处理程序以完成如下步骤:

根据所述标记位信息确定表示所述关系的的信息之表达式; 以及

15 把由所述第二表达式表示的所述字符串信息转换为由第一表达式表示的所述流 ID 信息, 其中如果表示所述关系的的信息是所述流 ID 信息, 则所述图像/音频数据按照所述的流 ID 信息与所述节点连接; 如果表示所述关系的的信息是所述字符串信息, 则所述图像/音频数据按照转换的流 ID 信息与所述节点连接。

40. 一种上面记录有数据处理程序的记录介质, 该数据处理程序用于从  
20 包括由多个节点构成的三维空间模型数据和图像/音频数据的比特流产生显示图像, 执行所述数据处理程序以完成如下步骤:

接收包括由多个节点构成的所述三维空间模型数据和图像/音频数据的所述比特流, 并输出所述三维空间模型数据和图像/音频数据;

25 把字符串信息转换为流 ID 信息, 所述字符串信息是表示节点与图像/音频数据之间的关系的的信息; 以及

按照转换的流 ID 信息连接所述图像/音频数据与所述节点。

41. 一种上面记录有包括由多个节点构成的三维空间模型数据和由所述三维空间模型数据的节点中包含的位置指定信息指定的图像/音频数据的比特流的记录介质, 所述比特流按如下步骤形成:

30 取出所述三维空间模型数据的节点中包含的位置指定信息;

把取出的位置指定信息转换为对应于由所取出的位置指定信息指定的

图像/音频数据的流 ID 信息;

用所述流 ID 信息替换所述节点中包含的位置指定信息; 以及

多路复用所述图像/音频数据和包含所述流 ID 数据的三维空间模型数据, 以便产生所述比特流。

- 5 42. 根据权利要求 41 的记录介质, 其中所述比特流进一步通过如下步骤形成:

把所述流 ID 信息转换为由第一表达式表示的字符串信息; 以及

- 10 确定是否用以第二表达式表示的所述流 ID 信息或以第一表达式表示的所述字符串信息替换所述节点中包含的所述位置指定信息, 其中根据确定的结果替换所述节点中包含的所述位置指定信息, 并将所确定的表达式的代表信息引入所述节点中代替被替换的信息。

43. 一种上面记录有包括由多个节点构成的三维空间模型数据和由在所述三维空间模型数据的节点中包含的位置指定信息指定的图像/音频数据的比特流的记录介质, 所述比特流按如下步骤形成:

- 15 取出所述三维空间模型数据的节点中包含的位置指定信息;

把取出的位置指定信息转换为对应于由所取出的位置指定信息指定的图像/音频数据的流 ID 信息;

把所述流 ID 信息转换为字符串信息;

- 20 用所述字符串信息替换所述节点中包含的所述位置指定信息; 以及  
多路复用所述图像/音频数据和包括所述字符串信息的三维空间模型数据, 以便产生所述比特流。

44. 一种由制造装置生产的记录介质, 所述记录介质上面记录具有数据流的信号, 所述数据流包括由多个节点构成的三维空间模型数据和图像/音频数据, 所述有记录信号的记录介质经过如下步骤处理:

- 25 接收包括由节点构成的所述三维空间模型数据和所述图像/音频数据的所述数据流;

输出所述三维空间模型数据和图像/音频数据; 以及

根据表示所述节点与所述图像/音频数据之间关系的信息连接所述图像/音频数据与节点, 表示所述关系的信息是流 ID 信息。

- 30 45. 根据权利要求 44 的记录介质, 其中表示所述关系的信息是由第一表达式表示的所述流 ID 信息和与所述流 ID 信息对应的字符串信息之一; 所述

字符串信息由第二表达式表示；而且其中所述节点包括标记位信息，该标记位信息标致表示所述关系之信息的表达式，所述有记录信号的记录介质进一步经如下步骤处理：

- 按照所述标记位信息确定表示所述关系的信息之表达式；以及
- 5 把由第二表达式表示的所述字符串信息转换为由第一表达式表示的所述流 ID 信息，其中如果表示所述关系的信息是所述流 ID 信息，则所述图像/音频数据按照所述的流 ID 信息与所述节点连接；如果表示所述关系的信息是所述字符串信息，则所述图像/音频数据按照所转换的流 ID 信息与所述节点连接。

- 10 46. 一种由制造装置生产的记录介质，所述记录介质上面记录具有流的信号，所述流包括由多个节点构成的三维空间模型数据和图像/音频数据，所述有记录信号的记录介质经过如下步骤处理：

接收包括由节点构成的所述三维空间模型数据和所述图像/音频数据的所述流；

- 15 输出所述三维空间模型数据和图像/音频数据；

把字符串信息转换为流 ID 信息，所述字符串信息是表示节点与图像/音频数据之间的关系的的信息；以及

根据被转换的流 ID 信息连接所述图像/音频数据与所述节点。

- 20 47. 一种用于提供由多个节点定义的三维空间模型数据和由各节点中包含的位置确定的图像/音频数据的设备，它包括：

用于从所述三维空间模型数据的节点中取出相应位置的分析电路；

把取出的位置转换为对应于与所述位置相连系之图像/音频数据的流 ID 的转换器；

用所述流 ID 替换所述位置的编码器；以及

- 25 用于多路复用所述图像/音频数据和包含所述流 ID 的所述三维空间模型数据的复用器，以便产生比特流。

48. 一种用于提供由多个节点定义的三维空间模型数据和由各节点中包含的位置确定的图像/音频数据的设备，它包括：

用于从所述三维空间模型数据的节点中取出相应位置的分析电路；

- 30 把取出的位置转换为对应于与所述位置相连系之图像/音频数据的流 ID 的转换器；

用于把所述流 ID 转换为字符串信息的变化电路;

用所述字符串替换所述位置的编码器; 以及

多路复用所述图像/音频数据和包含所述字符串的所述三维空间模型数据的多路复用器, 以便产生比特流。

- 5 49. 一种用于处理包括图像/音频数据和包含多个节点之三维空间模型数据的比特流, 以便产生显示图像的设备, 它包括:

用于接收所述比特流和把所接收的比特流多路分解为流 ID、所述三维空间模型数据和图像/音频数据的多路分解器; 以及

- 10 用于根据所述流 ID 提供所述图像/音频数据与各个节点之间的关系以便产生显示图像的重组电路。

50. 一种用于处理包括图像/音频数据和包含多个节点之三维空间模型数据的比特流, 以便产生显示图像的设备, 它包括:

用于接收包括图像/音频数据和由多个节点组成的所述三维空间模型数据的所述比特流并输出所述三维空间模型数据和图像/音频数据的多路分解

- 15 器;

用于把字符串信息转换为流 ID 信息的转换器, 所述字符串信息是表示节点与所述图像/音频数据之间关系的信息; 以及

用于根据所述转换的流 ID 信息连接所述图像/音频数据与所述节点的重组电路。

产生含有二进制  
图像/音频数据的比特流的  
方法和装置

5

本发明涉及用于在如光盘或磁带记录介质上记录移动图像信号和在显示装置上显示重放该信号的编码和解码装置与方法。本发明可以如下的方式用于电视会议系统、可视电话系统、广播设备、多媒体数据库检索系统以及类似系统，即移动图像信号通过传输线路从发射端传输到接收端并在接收端接收和显示。本发明还可以用于编辑和记录移动图像信号。

10

在把移动图像信号传输到远处的电视会议或可视电话系统中，为了有效地利用传输线，利用视频信号的行相关和帧相关把图像信号压缩/编码。近年来，随着计算机处理技术的发展，使用计算机的移动图像信息终端已经普及。在这样的系统中，通过例如网络的传输线把信息传输到远处。在这样的情况下，为了有效的利用传输线，将要传输的信号例如图像、声音或计算机数据在压缩/编码以后再传输。在终端(接收端)采用与编码方法相应的预定解码方法把传输来的压缩/编码信号解码为原始信号图像、声音或计算机数据，然后由显示装置、扬声器或类似的终端输出。以前，被传输的图像信号或类似信号仅仅在显示装置上以它实际的形式输出。但是在使用计算机的信息终端上，多个图像、声音或计算机数据在经过给定的转换过程以后可以在二维或三维空间处理或显示。可以如下方式进行这样的处理，即在发射端采用给定的方法描述二维或三维空间的信息，而终端(接收端)根据该描述方法对图像信号或类似信号进行转换处理。

15

20

25

VRML(Virtual Reality Modeling Language - 虚拟实现模型化语言)是描述空间信息的典型例子，已由 ISO - IEC/JTC1/SC24 将它标准化。最新的版本 VRML2.0 在 IS14772 中有所介绍。VRML 是描述三维空间的语言，它规定描述三维空间的属性、形状等的规定数据。这样的数据称为节点。为了描述三维空间，必须首先描述怎样组合这种节点。每个节点包括表示颜色、纹理等的数据、表示多边形形状的数据和其他信息。

30

在使用计算机的信息终端，利用 CG(计算机绘图)根据上述使用多边形等

- 的 VRML 生成给定的目标。利用 VRML 可以把一种纹理附加于以这样的方式生成的由多边形构成的三维目标上。称为“纹理(Texture)”的节点是为静止图像定义的,称为“影片纹理(Movie Texture)”的节点是为移动图像定义的。要附在纹理上的信息(文件名、显示开始时间或结束时间等等)在这些节点处
- 5 描述。将参考图 23 描述纹理附加过程(以下称为纹理绘制过程,是近似说法)。

- 图 23 示出了纹理映射绘制装置的结构例子。如图 23 所示,存储器组 200 包括纹理存储器 200a、灰度级存储器 200b 和三维目标存储器 200c。纹理存储器 200a 存储外部输入的纹理信息。灰度级存储器 200b 和三维目标存储器 200c 存储也是外部输入的代表纹理的穿透/透明程度的主要数据和三维
- 10 目标的信息。三维目标信息对产生多边形是必须的而且与照度有关。绘制电路 201 根据存储在存储器组 200 的三维目标存储器 200c 中的三维目标信息通过生成多边形产生三维目标。再者,根据三维目标数据,绘制电路 201 分别从存储器 200a 和 200b 中读出纹理信息和表示纹理的穿透/透明程度的主要数据,并且通过参考这些主要数据叠加到纹理和相应的背景图像上。主要数据
- 15 表示相应位置的纹理的穿透程度,即在相应位置的目标的透明度。

二维转换电路 202 根据外部输入的观察点的信息把二维图像信号输出到二维平面,所述二维图像信号通过变换由绘制电路 201 产生的三维目标获得。在纹理是移动图像的情况下,上述过程是在一帧一帧的基础上进行的。

- 通过 VRML,能够处理根据主要用于静止图像高效编码的 JPEG(Joint
- 20 Photographic Expert Group - 联合照相专家组)、用于移动图像高效编码的 MPEG(Moving Picture Expert Group - 移动图像专家组)等压缩的数据,作为纹理信息。当这样压缩的图像用作纹理时,通过与编码方案相应的解码处理解码该纹理(图像)。解码的图像存储在存储器组 200 的纹理存储器 200a 中并且经历与上面类似的处理。

- 25 绘制电路 201 把存储在纹理存储器 200a 中的纹理信息附加于给定位置的目标上,不管图像的格式和图像是否为移动的图片亦或是静止的图片。因此,可被附于某一多边形上的纹理存储在一个存储器中。在传输三维目标信息的过程中,必须传输每个顶点的三维坐标。每个坐标需要 32 位实数数据。反映每个三维目标的属性也需要 32 位或更多位的实数数据。因此,传输的信
- 30 息量很大而且在传输复杂的三维目标或移动图像时进一步增加。因此,在传输线传输上述的三维信息或纹理信息过程中,必须传输压缩的信息以便提高

传输效率。

5 高效编码(压缩)移动图像的方案的典型例子是 MPEG(Moving Picture Expert Group, 存储的编码移动图像)方案, 这在 ISO - IEC/JTC1/SC2/WG11 中有所讨论而且提出作为标准。MPEG 采用把移动补偿预测编码与 DCT(离散余弦变换)编码结合的混合方案。为了适用不同的应用和功能, MPEG 定义了几个分布(功能分类)和等级(例如图像大小的量)。最基本的形式是主分布(profile)的主等级(MP @ ML)。

10 MPEG 方案的 MP @ ML 编码器(图像信号的编码装置)结构的例子将参考附图 24 描述。首先向帧存储器 1 输入一个输入图像信号, 然后按预定的顺序编码。将要编码的图像数据根据宏模块输入到移动矢量检测电路(ME)2。移动矢量电路 2 根据预定的次序把每帧的图像数据作为 I - 图像、P - 图像或 B - 图像处理。也就是说, 预先确定顺次输入的各帧图像是否作为 I、P 或 B - 图像处理(例如, 按 I、B、P、B、P、...、B、P 的顺序处理)。

15 移动矢量检测电路 2 通过参考预定的参考帧进行移动补偿, 并检测它的移动矢量。移动补偿(中间帧预测)有三种预测模式, 即前向预测、后向预测和双向预测。只有前向预测适用于 P - 图像预测模式, 而三种模式预测, 即前向预测、后向预测和双向预测适用于 B - 图像预测模式。移动矢量检测电路 2 选择使得预测误差最小的预测模式并生成相应的预测矢量。

20 使产生的预测误差与例如将要编码的宏模块的变化量进行比较。如果宏模块的变化量小于预测误差, 不对宏模块进行预测而进行帧内编码。在这种情况下, 预测模式是图像内预测(内部预测)。将移动矢量检测电路 2 测得的移动矢量和上述的预测模式输入到可变量长度编码电路 6 和移动补偿电路(MC)12 中。移动补偿电路 12 根据给定的移动矢量生成预测图像数据并输入到工作电路 3 和 10 中。工作电路 3 计算表示所要编码的宏模块值和预测图像数据值之差的差值数据, 并把计算结果输出给 DCT 电路 4。在宏模块内模式情况下, 工作电路 3 把将要编码的宏模块数据按原样输出给 DCT 电路 4。

25 DCT 电路 4 使输入数据经 DCT(离散的余弦变换)处理变换为 DCT 系数。将该 DCT 系数输入到量化电路(Q)5, 在此经过量化级量化为与传输缓存器 7 的数据存储量(缓存器存储器)相应。量化的系数(数据)被输入到可变量长度编码电路 6。

30 可变量长度编码电路 6 把量化电路 5 输入的量化数据变换为例如 Huffman

码(霍夫曼码)的可变长度码。可变长度电路 6 也接收来自量化电路 5 和预测模式(表明设置了前向、后向和双向中的那种图像内预测)的量化阶(标度)以及来自移动矢量检测电路 2 的移动矢量, 然后在此进行可变长度编码。传输缓存器 7 暂时存储接收的编码数据并把与存储量相应的量化控制信号输出到量化电路 5。当剩余的数据量增加到允许的上限时, 传输缓存器 7 利用量化控制信号通过提高量化电路 5 的量化级控制降低量化数据的数据量。相反, 当剩余的数据量减少到允许的下限时, 传输缓存器 7 利用量化控制信号通过降低量化电路 5 的量化级控制增大量化数据的数据量。就以此方式防止了传输缓存器 7 上溢和下溢。存储在传输缓存器 7 中的编码数据按预定的读出并作为比特流输出到传输线。另一方面, 将从量化电路 5 输出的量化数据输入到逆-量化电路(IQ)8, 在此根据来自量化电路 5 的量化阶使之逆量化。逆量化电路 8 输出的数据(DCT 系数)输入到 IDCT(反 DCT)电路 9, 然后经过反 DCT 处理, 并经过工作电路 10 存储在帧存储器(FM)11 中。

下面, 将参考附图 25 描述 MPEG 的 MP @ ML 解码器(图像信号解码装置)的例子。经过传输线传输的编码图像数据(比特流)被接收电路(未示出)接收或者被再现电路再现, 暂时存储在接收缓存器 21 中, 然后提供给可变长度解码电路(IVLC)22。对来自接收缓存器 21 的数据进行可变长度解码, 可变长度解码电路 22 向移动补偿电路 27 输出移动矢量和预测模式, 并向逆量化电路 23 输出量化阶。而且, 可变长度解码电路 22 向逆-量化电路 23 输出解码量化数据。逆量化电路 23 根据仍由可为长度解码电路 22 提供的量化阶对可变长度解码电路 22 提供的量化数据进行逆量化, 并把所得数据(DCT 系数)输出给 IDCT 电路 24。逆量化电路 23 输出的数据(DCT 系数)在 IDCT 电路 24 经过反 DCT 变换并作为输出数据提供给工作电路 25。如果 IDCT 电路 24 提供的输出数据(输入比特流)是 I - 图像数据, 则它作为图像数据从工作电路 25 输出然后提供给帧存储器 26 并存储在那里, 用于产生图像数据(P 或 B - 图像)的预测图像数据, 所述图像数据将输入到工作电路 25。该图像数据也以原样作为再现图像输出到外部系统。

如果自 IDCT 电路 24 提供的输出数据(输入比特流)是 P 或 B - 图像, 那么移动补偿电路 27 在存储在帧存储器 26 中的图像数据的基础上, 根据可变长度解码电路 22 提供的移动矢量和预测模式生成预测图像, 并输出给工作电路 25。工作电路 25 把自 IDCT 电路 24 提供的输出数据和自移动补偿电路 27





提供的预测图像数据相加，得到输出图像数据。在 P - 图像的情况下，工作电路 25 的输出数据输入给帧存储器 26 并作为预测图像数据(参考图像)存储在那里，用于随后将要被解码的图像信号。

在 MPEG 中，定义了 MP @ ML 以外的不同分布和等级并准备了不同的工具。可量测性是这些工具之一。在 MPEG 中，引入了实现可量测性的可量测编码方案，以便适用于不同的图像大小和帧速率。例如，在空间可量测性的情况下，可以通过只解码低层比特流使图像尺寸小的图像信号被解码，而图像尺寸大的图像信号可以通过解码高层和低层比特流被解码。参考附图 26 描述空间可量测性的解码器。在空间可量测性情况下，所述低层对应于具有小的图像尺寸的图像信号，而所述高层对应于具有大的图像尺寸的图像信号。低层图像信号先被输入到帧存储器 1 中，再以与 MP @ ML 的情况相同的方式被解码。然而，工作电路 10 的输出不仅被提供给帧存储器 11，被用作低层预测图像数据，而且在由图像放大电路 31(采样)放大为与高层图像尺寸一样大小以后用作高层预测图像数据。根据附图 26，高层图像信号输入给帧存储器 51。移动矢量检测电路 52 以 MP @ ML 的情况相同的方式确定移动矢量和预测模式。移动补偿电路 62 根据移动矢量检测电路 52 确定的移动矢量和预测模式生成预测图像数据并输出给加权电路(W)34。加权电路 34 用加权因子 W 乘以预测图像数据并把加权的预测图像数据输出到工作电路 33。

如上所述，将工作电路 10 的输出数据(图像数据)输入到图像放大电路 31。图像放大电路 31 放大工作电路 10 生成的图像数据，使其大小等于高层图像尺寸，同时，图像放大电路 31 还把放大的图像数据输出给加权电路(1 - W)32。加权电路 32 用加权因子(1 - W)乘以图像放大电路 31 放大的图像数据，并把结果输出给工作电路 33。工作电路 33 把加权电路 32 和 34 的输出数据相加，并把结果输出给工作电路 53 作为预测图像数据。工作电路 33 的输出数据还被输入到工作电路 60，并在那里与反 DCT 电路 59 的输出数据相加，然后输入到帧存储器 61，以后用作要编码的图像数据的预测图像数据。工作电路 53 计算将要编码的图像数据的输出数据和工作电路 33 的输出数据之间的差值，并把作为差值数据的该结果输出。然而，在帧内编码宏块的情况下，工作电路 53 以原样把将要编码的图像数据输出给 DCT 电路 54。DCT 电路 54 对工作电路 53 的输出结果进行 DCT 变换(离散余弦变换)，

生成 DCT 系数，并输出给量化电路 55。与 MP @ ML 的情况一样，量化电路 55 根据量化阶量化 DCT 系数并把结果(量化数据)输出给可变长度电路 56，所述量化阶是以传输缓存器 57 的数据存储量及其他因素为基础。可变长度编码电路 56 对量化数据(量化的 DCT 系数)进行可变长度编码并把结果  
5 作为高层比特流经过传输缓存器 57 输出。量化电路 55 的输出数据被逆量化电路 58 用量化电路 55 中所用的量化阶逆量化，在反 DCT 电路 59 中经过反 DCT 变换，然后被输入到工作电路 60。工作电路 60 把工作电路 33 的输出和反 DCT 电路 59 的输出相加，并把结果输入到帧存储器 61。可变长度编码电路 56 还接收移动矢量检测电路 52 测得的移动矢量和预测模式、量化电路  
10 55 中所用的量化阶和加权电路 32 和 34 中所用的加权因子 W，这些值在可变长度编码电路 56 中编码，然后被传输。

下面，参考附图 27 描述空间可量测性解码器的例子。将低层比特流输入到接收缓存器 21，然后以与 MP @ ML 的情况相同的方式解码。然而，工作电路 25 的输出不仅被输出给外部系统并被存储在帧存储器 26 中用作以后  
15 要解码的图像信号的预测图像数据，而且在由图像信号放大电路 81 放大到与高层图像尺寸相同大小以后被用作高层预测图像数据。高层比特流通过接收缓存器 71 被提供给可变长度解码电路 72，并在此进行可变长度解码。也就是说，量化阶、移动矢量、预测模式和加权因子(W)与 DCT 系数一起被解码。经可变长度解码电路 72 解码的 DCT 系数(量化数据)由逆量化电路 73 利用  
20 解码的量化阶被逆量化，在反 DCT 电路 74 中经过反 DCT 变换，然后提供给工作电路 75。

移动补偿电路 77 根据解码的移动矢量和预测模式产生预测图像数据并输入给加权电路 84。加权电路 84 用解码的加权因子 W 乘以移动补偿电路 77 的输出，并把结果输出给工作电路 83。工作电路 25 的输出不仅作为低层  
25 再现图像信号输出给帧存储器 26，而且在由图像信号放大电路 81 放大以便与高层图像信号尺寸一样大小以后被输出给加权电路 82。加权电路 82 利用解码的加权因子 W 把图像信号放大电路 81 的输出乘以  $(1 - W)$ ，并把结果输出给工作电路 83。工作电路 83 把加权电路 82 和 84 的输出相加，并把结果输出给工作电路 75。工作电路 75 把反 DCT 电路 74 的输出和工作电路 83  
30 的输出相加，并把结果作为高层再现图像数据输出，同时还还将结果提供给帧存储器 76 用作以后要解码的图像数据的预测图像数据。

上面的描述可被用于照度信号的处理,以相似的方式可处理颜色差值信号,沿垂直方向和水平方向平分照度信号的移动矢量,获得颜色差值信号处理过程中所用的移动矢量。

- 虽然上面描述 MPEG 方案,但是用于移动图像的其他各种高效编码方案也已经标准化。例如,ITU - T(International Telecommunication Union - 国际电信联盟)已经把 H.261 和 H.263 方案标准化为通讯编码系统。基本上与 MPEG 方案一样,H.261 和 H.263 是移动补偿预测编码和 DCT 编码的结合。H.261 和 H.263 的编码装置和解码装置的结构与 MPEG 方案的一样,虽然标题信息等的细节不同。另外,在上述 MPEG 方案中,被称为 MPEG4 的新高效编码方案的标准化正在进行。MPEG4 的主要特点是,图像是在目标 - 目标的基础上编码的(一幅图像是按多幅图像为单元被编码的),而且可以在目标 - 目标的基础上使图像得到调整。也就是说,在解码端,可将各个目标的图像或者多个图像合成,重组为一幅图像。

- 如前所述,在 ISO - IEC/JTC1/SC29/WG11 中,MPEG4 的标准化工作正在进行。在这一工作中,正在研究通常体制中处理自然图像和计算机绘制图像的方案。在这一方案中,利用 VRML 描述三维目标,根据 MPEG 标准压缩移动的图像和声音或音频信号。根据 VRML 描述由多个三维目标、移动的图像等构成的场景。对场景的描述(后面简称为场景描述)、三维目标的描述和由根据 MPEG 方案压缩之移动图像、声音或音频信号构成的 AV 数据(以 20 上述方式获得的)给以时间标记,并由多路传输电路将其多路复用为比特流,作为多路传输比特流传输。在接收多路传输比特流的接收端,多路传输电路取出场景描述、三维目标描述和 AV 流(与 AV 数据相应的流),多个解码器解码各个比特流,并将由场景重组电路重组的场景显示在显示装置上。

- 在上面的方法中,必须搞清楚根据 VRML 描述的各节点(三维目标描述 25 和场景描述)与移动图像、声音、音频信号等的 AV 数据之间的关系。例如,必须表明应使怎样的 AV 流用某一三维目标进行纹理绘制。在 VRML 中,由 URL(Uniform Resource Locator,这是表示网络上的服务器的字符串)指定拟被附于(被绘以)三维目标的纹理。这种指定方法与网络上的 AV 数据文件的绝对地址的指定相应。另一方面,在根据 MPEG 方案的系统中,通过指定每种 30 AV 流的 ID 来识别它。这与当建立起对话(通讯线路)时一种流在对话中的相对路径的指定相应。也就是说,在 VRML 中,除了使用 URL 以外没有其他



方法用于识别流。但是，例如使用 MPEG 实时通讯需要基于 ID 的识别。在两个方案之间存在不匹配的问题。

- 从另一点看，可以说 VRML 设定了一种客户需要信息的模式。另一方面，MPEG 设定了一种广播信息或类似信息在服务器控制下传输的模式。这些模式之间的差别导致了如下问题，即虽然与 VRML2.0 兼容，但是难以将计算机绘制的图像与自然图像融合在一起。

本发明是在上述情况下提出的，因此本发明的目的是使根据 VRML 描述的计算机绘制图像和根据 MPEG 方案压缩的图像或类似信息能够按如下状态传输，即它们被多路复用于同一比特(数据)流中。

- 在产生由多个节点定义的三维空间模型数据和由包括所述节点位置描述的图像/音频数据的方法中，进行如下步骤：从三维空间模型数据的节点中取出相应的位置；把取出的位置转换为对应于与该位置相连系的图像/音频数据的流 ID；用流 ID 替换所述位置；以及多路传输所述图像/音频数据和包括所述流 ID 的三维空间模型数据，产生比特流。

- 根据本发明的一个方面，用 Virtual Reality Modeling Language(VRML - 虚拟现实模型化语言)描述三维空间模型数据，位置采用 ASCII 格式的 Uniform Resource Locator(URL - 统一资源定位符)表示，流 ID 用二进制格式表示。

- 根据本发明的另一方面，将流 ID 转换为字符串，而且根据图像/音频数据是由一个服务器提供还是由多个服务器提供确定是用流 ID 还是用字符串替换图像/音频数据的位置。

通过下面详细描述目前的最佳实施例，本发明的其他目的和优点将更加明了，下面的描述是结合附图进行的，其中：

- 图 1 是表示本发明编码装置第一实施例结构实例的方块图；
- 图 2 表示场景描述 SD 与节点之间的关系；
- 图 3 表示移动图像作为纹理附加于节点用的场景描述的 ASCII 格式的实例；
- 图 4 表示静止图像作为纹理附加于节点用的场景描述的 ASCII 格式的实例；
- 图 5 表示移动图像作为纹理附加于节点用的场景描述的二进制格式的实例；

图6表示静止图像作为纹理附加于节点用的场景描述的二进制格式的实例;

图7表示图1所示多路传输装置详细结构的实例;

图8是表示本发明解码装置第一实施例结构实例的方块图;

5 图9表示图8所示的多路分解电路404详细结构的实例;

图10表示图8所示重组电路411结构的实例;

图11是表示图9所示合成电路详细结构实例的方块图;

图12示出目标描述器OD的实例;

图13示出“ES - Descriptor”的实例;

10 图14示出“ES - ConfigParams”的实例;

图15是表示本发明编码装置第二实施例结构实例的方块图;

图16表示移动图像作为纹理附加于节点用的场景描述的二进制格式实例;

15 图17表示静止图像作为纹理附加于节点用的场景描述的二进制格式实例;

图18是表示本发明解码装置第二实施例结构实例的方块图;

图19是表示本发明编码装置第三实施例结构实例的方块图;

图20表示移动图像作为纹理附加用的场景描述SD的二进制格式实例;

20 图21表示静止图像作为纹理附加用的场景描述SD的二进制格式实例;

图22是表示本发明解码装置第三实施例结构实例的方块图;

图23是纹理绘制的方块图;

图24是表示MPEG方案的一个MP @ ML编码器实例的方块图;

图25是表示MPEG方案的一个MP @ ML解码器实例的方块图;

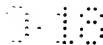
图26是表示空间可量测性的编码器实例的方块图;

25 图27是表示空间可量测性的解码器实例的方块图;

下面参考附图详细描述本发明的最佳实施例。

图1是本发明编码装置第一实施例结构实例的方块图。

参见图1, 系统控制电路301接收请求信号(请求(QER)), 通过参考存储在存储装置302中的场景描述SD(下面将详细描述)确定应该传输什么样的  
30 AV目标(三维目标、自然图像、声音等), 并向存储装置302输出场景请求信号(场景请求(SQER))。存储装置302存储描述二维或三维场景的场景描述



SD. 根据遵守 VRML2.0 的 ASCII 格式描述场景描述 SD. 存储装置 306 存储音频和视频(AV)数据比特流(基元流(ES)), 例如移动图像、静止图像和声音. 存储装置 305 存储为了解码存储装置 306 中所存的 AV 目标所必需的信息(目标流信息(OI)). 例如, 目标流信息 OI 是解码 AV 目标所必需的缓存器大小或者每个访问单元的时间标记. 目标流信息 OI 包括与各个 AV 目标对应的所有 AV 比特流信息.

下面将参考图 2 描述场景描述、AV 数据(流)和三维目标之间的关系. 在图 2 的例子中, 计算机绘图生成的长方形图像序列和三角锥体显示于场景 352 中. 虽然在这一例子中没有纹理附加于三角锥体上, 但是正象其他三维目标的情况一样可将纹理附加于其上. 所加的纹理可以是静止图像或移动图像.

场景描述 SD350 包括称为节点的描述. 存在一个本源(根)节点 SD0, 描述在整幅图像中怎样排列目标. 节点 SD1 是本源节点 SD0 的子节点, 描述与三角锥体相关的信息. 节点 SD2 也是本源节点 SD0 的子节点, 描述与要附加上图像的长方形平面相关的信息.

在图 2 中, 图像信号包括三个视频目标 VO(背景、太阳和人). 节点 SD2 描述与背景相关的信息. 节点 SD3 描述与附加有太阳的长方形平面相关的信息. 节点 SD4 描述与附加人的平面相关的信息. 每个节点描述表示相应的 AV 数据(比特流)文件的地址的 URL. 节点 SD3 和 SD4 是节点 SD2 的子节点.

一个的场景描述 SD 是所有节点 SD0 - SD4 的集合. 下面将所有节点描述的集合称为场景描述, 而将各个节点称为目标(二维或三维目标). 因此, 每个节点对应于单独一个二维物或三维目标. 每个目标一一对应于一个目标描述符 OD, 它描述与该目标相关的 AV 数据(比特流).

参见图 1, 对分析电路 307 读出在节点处描述的由存储装置 302 输出的 URL(表示 AV 文件的地址), 并向存储装置 306 输出用于请求输出与 URL 相应的 AV 数据(比特流)的请求信号(SE 请求(ESREQ)). 再有, 分析电路 307 向存储装置 305 输出用于请求输出目标流信息 OI 的请求信号(OI 请求(OIREQ)), 所述目标流信息 OI 描述与所述 URL 相应的 AV 数据(比特流)的相关信息.

OD(目标描述符)生成电路 304 接收与存储装置 305 输出的 AV 目标相关的目标流信息 OI, 并仅取出由请求信号 OIREQ 请求的 AV 数据(比特流)信

息，作为目标描述符 OD，输出给多路复用电路 303。而且，OD 生成电路 304 为每个取出的目标描述符 OD 生成一个 ID 编号 OD - ID，记录在目标描述符 OD 中，并把所得的目标描述符 OD 输出给多路复用电路 303，而且还把生成的 ID 编号 ID - OD 输出给 BIFS 编码器 308。

- 5 BIFS 编码器 308 把从存储装置 302 输出的 ASCII 格式的场景描述转换为二进制格式，并用 OD 生成电路 304 输出的 ID 编号 OD - ID 代替包括在场景描述 SD 中的 URL。然后，BIFS 编码器 308 把已经转换为二进制格式并用 ID 编号 OD - ID 代替的场景描述 B - SD 输出给多路复用电路 303。

- 10 多路复用电路 303 按规定的顺序多路复用存储装置 306 中所存的 AV 数据(比特流)、已经由 BIFS 编码器 308 转换为二进制的场景描述 B - SD、以及由 OD 生成电路 304 生成的目标描述符 OD，并把多路传输结果作为多路传输比特流 FS 输出。后面将参考图 7 描述多路复用电路 303 的详细例子。

- 下面描述上述实施例的工作情况，当用户从外部终端(未示出)输入产生所要显示的某一 AV 目标的请求信号时，请求信号(REQ)提供给系统控制电路 15 301。系统控制电路 301 一旦收到请求信号 REQ，就根据请求信号 REQ 参考存储装置 302 中所存的场景描述 SD 确定应该传输什么样的 AV 目标，并向存储装置 302 输出场景请求信号 SREQ。存储装置 302 一旦收到场景请求信号 SREQ 就读出相应的场景描述 SD(以 ASCII 格式描述)，并提供给分析电路 307 和 BIFS 编码器 308。

- 20 图 3 表示移动图像作为纹理附加用的场景描述(以 ASCII 格式描述)的实例。在这个例子中，第六行描述表示将要附加的移动图像文件的地址的 URL。图 4 表示静止图像作为纹理附加用的场景描述(以 ASCII 格式描述)的实例。在这个例子中，第二行描述表示将要附加的静止图像文件的地址的 URL。图 3 和图 4 的格式遵守 VRML 的节点描述。

- 25 分析电路 307 读出构成所提供的场景描述 SD 的节点中包括的 URL(表示 AV 数据(比特流)文件的地址)，并向存储装置 306 输出请求信号 ESREQ。结果，从存储装置 306 中输出相应的 AV 数据(比物流)，并将之提供给多路复用电路 303。

- 30 另外，分析电路 307 向存储装置 305 输出用于请求输出目标流信息 OI 的请求信号 OIREQ，所述目标流信息 OI 与由所述节点中所包括的 URL 表示的 AV 数据(比特流)ES 相关。结果，与 URL 相应的目标流信息 IO 从存储



装置 305 输出给 OD 生成电路 304。

- OD 生成电路 304 仅取出由来自与存储装置 305 提供的 AV 目标相关的目标流信息 OI 的请求信号 OIREQ 请求的信息, 作为目标描述符 OD。而且, OD 生成电路 304 生成一个 ID 编号 ID - OD, 将其记录在目标描述符 OD 中, 并把所得的目标描述符 OD 输出给多路复用电路 303。此外, OD 生成电路 304 还把为每个目标描述符 OD 生成的 ID 编号 OD - ID 输出给 BIFS 编码器 308。

- BIFS 编码器 308 把从存储电路 302 输出的 ASCII 格式的场景描述 SD 采用预定的方法转换为二进制格式数据(场景描述 B - SD), 并替换场景描述 SD 中所包括的 URL。然后, BIFS 编码器 308 把已经转换为二进制格式的场景描述 B - SD 输出给多路复用电路 303。在已经由 ISO 标准化被称为 MPEG4WD 的文件(文件号 N1825)中详细描述了二进制格式。下面将描述二进制格式的例子。

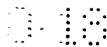
- 图 5 表示把附加移动图像作为纹理所用的场景描述(ASCII 格式, 见图 3)变换为二进制格式所得的数据。在图 5 中, 出现在 29 行上的“目标描述符 ID”是表示拟被附加于该节点之移动图像的 ID 编号的 OD - ID 标记。BIFS 编码器 308 把由 OD 生成电路 304 提供的 ID 编号 OD - ID 写在已经转换为二进制格式之场景描述 B - SD 中的这一部分。结果, 以 ASCII 格式描述为 URL 的 AV 数据(比特流)的地址转换为 ID 编号 OD - ID(二进制格式)。

- 图 6 表示把附加静止图像作为纹理所用的场景描述(ASCII 格式, 见图 4)变换为二进制格式所得的数据。在这一例子中, 出现在 17 行上的“目标描述符 ID”和 ID 编号的 OD - ID 被写在已经转换为二进制格式之场景描述 B - SD 中的这一部分。将这样生成的二进制格式场景描述 B - SD 提供给多路复用电路 303。多路复用电路 303 按照规定的顺序多路传输存储装置 306 中所存的 AV 数据(比特流)、已由 BIFS 编码器 308 转换为二进制的场景描述 B - SD、以及由 OD 生成电路 304 生成的目标描述 OD, 并输出多路复用比特流 FS。

图 7 表示多路复用电路 303 之详细结构的实例。在图 7 中, 开始代码生成电路 303a 生成并输出表示比特流开始位置的开始代码。

- 存储装置 306 输出的 AV 数据(比特流)ES - ESN 提供给相应的终端。BIFS 编码器 308 输出的二进制格式场景描述 B - SD 和 OD 生成电路 304 输





出的目标描述符 OD 提供给相应的终端。另外，开始代码生成电路 303a 输出的开始代码提供给相应的终端。

5 多路复用电路 303 操纵一个开关，以便连接与开始代码生成电路 303a 相连的终端，从而输出开始代码。然后，切换到使场景描述 SD 被输入的终端，从而输出场景描述 SD。再后，切换到使目标描述符 OD 被输入的终端，从而输出目标描述符 OD。最后，根据数据顺次切换到使 AV 数据(比特流)被输入的终端，从而输出 AV 数据(比特流)ES1 - ESN。

10 多路复用电路 303 通过开关选择开始代码、场景描述 SD、目标描述符 OD、和 AV 数据(比特流)，从而作为多路复用的比特流 FS 向外部系统输出这些信息。例如，通过传输线把多路复用比特流 FS 提供给接收终端。

下面参考图 8 描述与图 1 中编码装置相应的解码装置的一个实施例结构的实例。图 8 是表示根据本发明的解码装置的一个实施例结构实例的方块图。在图 8，多路分解电路 404 接收多路复用比特流 FS，然后分离并取出构成多路复用比特流 FS 的各个比特流。

15 图 9 表示多路分解电路 404 结构的实例。如图 9 所示，多路分解电路 404 检测多路复用比特流 FS 中的开始代码并识别各个比特流的存在。然后，通过开关把输入的多路复用比特流 FS 分离为场景描述 SD 和目标描述符 OD，并从相应的终端输出。类似地，AV 数据的比特流 ES1 - ESN 被分离并从相应的终端输出。

20 再参见图 8，分析电路 406 接收由多路分解电路 404 分离的目标描述符 OD，确定解码 AV 数据(比特流)所必需的解码器类型和数目，并生成提供给相应解码器的各个 AV 数据(比特流)的比特流。而且，分析电路 406 从目标描述符 OD 中读出解码各比特流所需的缓存器容量，并把它们(Init)提供给各解码器 407 - 409。更且，为了能够确定各个比特流 ES1 - ESN 属于哪个节点，  
25 分析电路 406 向解码各目标描述 OD 中所描述的比特流的解码器输出各目标描述的 ID 编号 OD - ID。

解码器 407 - 409 根据预定的解码方法与编码方法对应地解码比特流，并向重组电路 411 输出生成的视频数据或音频/声音数据。另外，解码器 407 - 409 向重组电路 411 输出表示各个解码数据(视频数据或音频(声音)数据)属于哪个节点的 ID 编号 OD - ID。更且，如果接收到的比特流是表示图像大小和显示位置的数据(SZ, POS)以及包括在比特流中表示图像穿透程度

30

的数据(主要数据), 解码器 407 - 409 从比特流中解码表示图像大小和显示位置的数据(SZ, POS)(图像的大小和显示位置数据)以及表示图像穿透程度的数据(主要数据), 并把这些数据输出给重组电路 411。

虽然在上面的实施例对于 N 等于 1 至 3 的情况提供了三个解码器 407 - 409, 应该明白, 可以根据拟处理的数据改变解码器的数目。

分析电路 410 分析二进制格式的场景描述 B - SD 并把产生的数据提供给重组电路 411。另外, 分析电路 410 读取与目标描述中的 ID 编号 OD - ID 相对应的场景描述 B - SD 中的 ID 编号 OD - ID, 并提供给重组电路 411。

图 10 示出了再现整个图像的所有比特流之间的关系和一个重组电路 411 实例。如图 10 所示, 重组电路 411 包括合成电路 351; 合成电路 351 产生的图像信号提供给显示装置 352, 从而使图像在此显示。在图 10 中, 合成电路 351 和显示装置 352 被表示为重组电路 411。这是为了表示在合成电路 351 中产生的图像是怎样在显示装置 352 显示的。实际上显示装置 352 不包括在重组电路 411 中。合成电路 351 接收分析电路 410 提供的节点数据和 ID 编号 OD - ID, 以及解码器 407 - 409 提供的图像数据、主要数据、图像大小和显示位置信息(SZ, POS)和 ID 编号 OD - ID, 并搜索与 OD - ID 相应的图像数据, 并根据主要数据及大小和显示位置信息把图像数据附加于节点, 并把与生成的图像数据相应的图像信号输出给显示装置 352。

图 11 是表示重组电路 411 实例的方块图。如图 11 所示, 重组电路 411 包括匹配电路 360、目标合成电路 500 - 502 和二维转换电路 503。目标合成电路 500 包括存储器组 500 - 1 和再现电路 500 - 2。存储器组 500 - 1 包括纹理存储器 500 - 1a、灰度级存储器 500 - 1b 和三维目标存储器 500 - 1c。

例如, 纹理存储器 500 - 1a 存储解码器 407 提供的 AV 数据(比特流), 作为纹理数据。灰度级存储器 500 - 1b 存储解码器 407 提供的表示穿透程度的主要数据。三维目标存储器 500 - 1c 存储分析电路 410 输出的三维目标的信息(节点)。三维目标信息(节点)包括形成多边形的信息、照明多边形的照度信息及其他信息。图像的大小和显示位置数据(SZ, POS)也存储在例如灰度级存储器 500 - 1b 的某一位置。

再现电路 500 - 2 根据三维物存储器 500 - 1c 中所存的节点信息利用多边形产生三维目标。另外, 再现电路 500 - 2 接收分别从纹理存储器 500

1a 和灰度级存储器 500 - 1b 接收纹理数据和表示穿透程度的主要数据, 并且把纹理附加于相应的节点, 执行与主要数据相应的处理, 使纹理具有预选的透明度。将这样获得的数据输出给二维转换电路 503。另外, 将图像大小和显示位置数据(SZ, POS)输出给二维转换电路 503。因为目标合成电路 501 和 502 按与目标合成电路 500 相同的方式设计, 所以这里不再描述它们。如果把纹理(图像数据)附加于目标(绘制), 需要清楚纹理与目标之间的关系。为了搞清楚这种关系, 使用在目标描述符 OD 中描述的 ID 编号 OD - ID 和在场景描述 B - SD 中描述的 ID 编号 OD - ID。这样, 输出给分析电路 411 的数据在提供给相应的目标合成电路 500 - 502 之前, 首先提供给匹配电路 360。匹配电路 360 把在目标描述符 OD 中描述的 ID 编号 OD - ID 与在场景描述 B - SD 中描述的 ID 编号 OD - ID 匹配, 如图 8 所示, 从而找出相互关系。

二维转换电路 503 根据外部输入的观察点信息及目标合成电路提供的图像大小和显示位置数据, 通过绘制成二维平面, 把从各目标合成电路 500 - 502 输出附加纹理的目标转换为二维图像信号。将所得的二维图像信号提供给显示装置 352, 以在上面显示。

下面参考图 8 描述上述实施例的工作情况。将通过传输线传输的多路复用比特流 FS 提供给多路分解电路 404。多路分解电路 404 检测多路复用比特流 FS 中的开始代码并识别各比特流。然后, 多路分解电路 404 通过适当地转换图 9 所示的开关从多路复用比特流 FS 中分离出场景描述 B - SD、目标描述符 OD、与 AV 数据(比特流)相应的比特流 ES1 - ESN, 并输出这些数据。将目标描述符 OD 提供给分析电路 406, 将比特流 ES1 - ESN 提供给各个的解码器 407 - 409, 而将二进制格式的场景描述 B - SD 提供给分析电路 410。

分析电路 410 分析多路分解电路 404 输出的二进制格式场景描述 B - SD, 并把结果(三维物信息(NODE))提供给重组电路 411。另外, 分析电路 410 解码附加于节点的 AV 数据(比特流)的目标描述 OD 的 ID 编号 OD - ID, 并将它们提供给重组电路 411。

分析电路 406 接收目标描述符 OD, 确定解码比特流所需的解码器类型和数目, 并生成拟提供给各解码器的比特流。而且, 分析电路 406 从目标描述符 OD 中读出解码各比特流所需的缓冲器的容量或者每个访问单元的时间

标记，并把它们作为初始化信息(Init)提供给各解码器 407 - 409。于是，解码器 407 - 409 参考所提供的值(初始化信息(Init))完成初始化。另外，为了表示各解码器 407 - 409 处理的比特流属于什么目标，分析电路 406 输出各个目标描述符 OD 的 ID 编号 OD - ID。

- 5 解码器 407 - 409 根据分析电路 406 提供的初始化信息完成初始化，例如确定缓存器。当接收到与多路分解电路 404 输出的 AV 数据(比特流)相应的比特流时，解码器 407 - 409 采用与编码操作相应的预定解码方法解码各个比特流，并向再现电路 41 路输出生成的视频数据或音频(声音)数据。

- 10 进一步，解码器 407 - 409 向重组电路 411 输出表示已由各个解码器解码的比特流对应什么目标的 ID 编号 OD - ID。更且，如果解码的比特流是图像，解码器 407 - 409 输出表示图像的大小和显示位置的数据(SZ, POS)以及表示图像的穿透程度的数据(主要数据)。

- 如图 11 所示，将输出给重组电路 411 的数据提供给相应的目标合成电路 500 - 502。一个目标合成电路对应每个节点。如上所述，当把不同类型的  
15 的数据提供给相应的目标合成电路 500 - 502 时，必须知道各个解码器 407 - 409 处理的比特流属于什么目标。因此，在把数据提供给相应的目标合成电路之前，由匹配电路 360 使在目标描述 OD 中描述的 ID 编号 OD - ID 与在场景描述 B - SD 中描述的 ID 编号 OD - ID 进行核对。从而，能够找出所确定的信号(比特流)与三维目标信息(NODE)之间的关系。目标合成电路  
20 500 - 502 接收包含分别由解码器 407 - 409 的节点所表示的 ID 编号 OD - ID 的解码信号。如果接收的解码信号是图像数据，目标合成电路 500 - 502 就把该图像附加到所要生成的二维或三维目标上。

- 下面将对作为例子用的目标合成电路 500 描述上述工作情况。把要附加于目标上的纹理数据存储于纹理存储器 500 - 1a 中。将主要数据和 ID 编号  
25 OD - ID 提供给灰度级存储器 500 - 1b 并存储在这里。将节点(三维物信息)存储在三维目标存储器 500 - 1c 中。另外，将图像的大小和显示位置数据(SZ, POS)也存储在例如灰度级存储器 500 - 1b 的某一位置。ID 编号 OD - ID 用于识别节点。

- 再现电路 500 - 2 读取三维目标存储器 500 - 1c 中所存的节点(三维目标信息)并利用多边形生成相应的目标。另外，再现电路 500 - 2 参照从灰度级存储器 500 - 1b 接收的表示穿透程度的主要数据，把从纹理存储器 500



- 1a 接收的图像数据附加于上述生成的多边形上。并且，从灰度级存储器 500 - 1b 读取图像大小和显示位置数据(SZ, POS)并提供给二维转换电路 503。目标合成电路 501 和 502 进行类似的工作过程。

将来自各个景物合成电路 500 - 502 的附加纹理的二维或三维目标提供  
5 给二维转换电路 503。根据外部输入的观察点的信息及图像大小和显示位置数据(SZ, POS)，二维转换电路 503 通过绘制二维平面把三维目标转换为二维图像信号。已经转换为二维图像信号的三维目标在显示装置 352 上面输出(显示)。

如果所有的目标都是二维的，各个再现电路 500 - 2 至 502 - 2 的输出  
10 被按它们的穿透程度(主要数据)原样合成然后输出。在这种情况下，不进行转换。

图 12 - 14 示出目标描述 OD 的结构。图 12 示出目标描述 OD 的整体  
结构。在图 12 中，第三行的“NodeId”是表示这一目标描述的 ID 编号的  
10 10 位标记位，而且与上述 ID 编号 OD - ID 相对应。第四行的术语  
15 “streamCount”是表示包括在目标描述符 OD 中的 AV 数据单元(比特流 ES)  
数目的 8 位标记位。因此，按照“streamCount”表示的数字传输解码各个  
比特流 ES 所必须的术语“ES - Descriptor”。第五行的术语“extensionFlag”  
是表示是否传输其他信息的标记位。如果这一标记位的值为“1”，则传输其  
他描述。

20 第八行的“ES - Descriptor”是表示与每个比特流相关的信息的描述。  
图 13 示出“ES - Descriptor”的细节。在图 13 中，第三行的“ES - number”  
是表示用于比特流识别的 ID 编号的 5 位标记位。第六行的术语  
“StreamType”表示比特流的格式，例如是表示如 MPEG2 视频数据的 8 位  
标记位。术语“Qos - Descriptor”是表示在传输过程中向网络请求的 8 位标  
25 记位。第八行的术语“ESConfigParams”是描述解码比特流所必须的信息的  
描述，图 14 中示出其细节。“ESConfigParams”的细节在 MPEG4 系统中有  
所描述。

在上述实施例中，在解码装置中，包括在构成三维空间模型数据(VRML  
数据)的节点中的 URL 被与由 URL 指定的 AV 数据(比特流)相对应的目标描  
30 述符 OD 的 ID 编号 OD - ID 代替。在解码端，搜索(排序)与节点中包括的  
ID 编号 OD - ID 相对应的目标描述符 OD，从而检测(识别)出相应的 AV 数



据(比特流)。因此,能够传输多路复用在同一比特流中的 CG 图像和自然图像,同时描述场景和三维目标的方法保持与例如 VRML 方案兼容。

在上述实施例中编码的音频和视频数据(AV 数据(比特流))被存储在存储装置 306 中。然而,例如可以直接从音频或视频编码装置输入这些数据而不

5 通过这样的存储装置。

虽然上面的实施例中 AV 数据(比特流)、目标描述符 OD 和场景描述 SD 存储在分开的存储装置中,但它们可以存储在同一存储装置或记录介质中。

而且,虽然场景描述 SD 预先以文件方式存储,但是 AV 数据(比特流)和目标流信息 OI 可在传输时实时生成。

下面参考附图 15,描述根据本发明的第二实施例的编码装置。在图 15 中,与图 1 中相对应的部分在后面采用相同的参考标记并不再描述。

本实施例中,在图 1 的实施例中加入一个 URL 变化电路 309。将分析电路 307 的输出数据和 OD 生成电路 304 的输出提供给 URL 变化电路 309,再将 URL 变化电路 309 的输出数据提供给 BIFS 编码器 308。其余的结构与图 1 中的实施例相同。

URL 变化电路 309 把 OD 生成电路 304 输出的 ID 编号 OD - ID 变换为相应的 ASCII 码格式的字符串,然后输出,例如,给出一个描述的例子,其中拟被附加于存储装置 302 中所存的某一节点的解码 AV 数据(比特流)所需的目标流信息 OI 具有如下地址:

http: //serverA/AV - scene1/object - file.1 .....(1)

在这种情况下,从存储装置 305 中读出目标流信息 OI,从 OD 生成电路 304 提供与目标流信息 OI 相应的目标描述符 OD 的 ID 编号 OD - ID。URL 变化电路 309 接收 ID 编号 OD - ID 并把 URL 改写(变化)为适当的 ASCII 格式的字符串。例如,如果 OD - ID 是“4”,表达式 1 改写(变化)为如下形式:

mpeg4: //4 .....(2)

其中字符串“mpeg”位于表示 URL 的字符串的头部,而且紧跟在位于“mpeg”之后的字符串“//”后面的表示数字的字符串(本例中的字符“4”)表示 ID 编号 OD - ID。

可能存在如下情况,即以存储装置 302 中所存节点描述的 URL 指定一



个存在于与图 15 所示编码装置不同的编码装置(在网络上)上的文件。在这种情况下, URL 变化电路 309 停止变化操作, 而且把表达式(1)的 URL 例如以原样提供给 BIFS 编码器 308。

- 下面简要描述本实施例的工作情况。当接收到请求信号 REQ 时, 场景
- 5 控制电路 301 参考存储在存储装置 302 中的场景描述 SD 根据请求信号 REQ 确定应该传输什么 AV 物, 并且向存储装置 302 输出场景请求信号 SREQ。

在接收场景请求信号 SREQ 的时候, 存储装置 302 读取相应的场景描述 SD(按 ASCII 格式描述的)并提供给分析电路 307 和 BIFS 编码器 308。

- 分析电路 307 读出包括在构成所提供之场景描述 SD 的节点中的 URL(表
- 10 示 AV 数据(比特流)文件的地址), 并向存储装置 306 输出请求信号 ESREQ, 用于输出与 URL 相应的 AV 数据(比特流)。结果, 相应的 AV 数据(比特流)ES 从存储装置 306 输出并提供给多路复用电路 303。

- 另外, 分析电路 307 向存储装置 305 输出用于请求输出目标流信息 OI 的请求信号 OIREQ, 所述目标流信息 OI 与由包括在所述节点中的 URL 表
- 15 示的 AV 数据(比特流)ES 相关。结果, 与 URL 相应的目标流信息 OI 从存储装置 305 输出, 并提供给 OD 生成电路 304。再有, 分析电路 307 把包括在节点的 URL 输出给 URL 变化电路 309。

- OD 生成电路 304 从与存储装置 305 提供的 AV 目标相关的目标流信息 OI 中仅取出由请求信号 OIREQ 请求的目标流信息, 作为目标描述符 OD。
- 20 而且, OD 生成电路 304 生成一个 ID 编号 OD - ID, 记录在目标描述符 OD 中, 并把所得的目标描述符 OD 输出给多路传输电路 303。再有, OD 生成电路 304 还把为每个目标描述 OD 生成的 ID 编号 ID - OD 输出给 URL 变化电路 309。

- 如果从分析电路 307 提供的 URL 指定一个存在于网络中的另一个服务
- 25 器上的文件, URL 变化电路 309 按照原样把 URL 输出给 BIFS 编码器 308。如果所提供的 URL 指定一个存储在存储装置 306 中的 AV 数据(比特流)文件, URL 变化电路 309 参照 OD 生成电路 304 输出的 ID 编号 OD - ID 生成一个如同表达式(2)的字符串, 并将其输出给 BIFS 编码器 308。

- BIFS 编码器 308 把从存储装置 302 提供的 ASCII 格式的场景描述 SD
- 30 用预定的方法转换为二进制格式的场景描述 B - SD, 并用 OD 生成电路 304 提供的 URL 或字符串替换包括在场景描述 SD 中的 URL。然后, 二进制格

式的场景描述 B - SD 被输出给多路复用电路 303。

图 16 表示按照作为纹理附加移动图像的二进制格式的场景描述 SD 的实例。在第 29 行上的 URL 是从 URL 变化电路 309 输出的 ASCII 格式的字符串。也就是说，在本实施例中，URL 被描述为按二进制形式的字符串。

- 5 图 17 表示按照作为纹理附加静止图像的二进制格式的场景描述的实例。如同图 16 的情况一样，在图 17 的第 17 行上的 URL 是 ASCII 格式的字符串。

- 已经由 BIFS 编码器 308 转换为二进制的场景描述符 SD 提供给多路复用电路 303，而且与目标描述 OD 和 AV 数据(比特流)ES 多路复用。所得的多路复用比特流 FS 从多路复用电路 303 输出。多路复用比特流 FS 通过例如传输线提供给解码装置。

下面参考附图 18 描述与图 15 的编码装置相应的解码装置的一个实施例。图 18 是表示根据本发明的解码装置第二实施例结构实例的方块图。在图 18 中，与图 18 中相对应的部分在后面采用相同的参考标记并不再描述。

- 15 图 18 的实施例中，是在图 8 的实施例中加入一个 URL 变化电路 412。另外，分析电路 410 提供作为 ASCII 格式的字符串所表示的信息。其余的结构与图 8 中的实施例相同。URL 变化电路 412 把以 ASCII 格式的字符串所表示的信息转换为 ID 编号 OD - ID，并提供给重组电路 411，ID 编号 OD - ID 为相应的目标描述符 OD 的 ID。

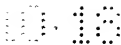
- 20 下面简要描述本实施例的工作情况。把由分析电路 410 从节点提取出的 URL 提供给 URL 变化电路 412。如果所述 URL 是具有例如表达式(2)的格式的字符串，则 URL 变化电路 412 把字符串变换为 ID 编号 OD - ID，并提供给重组电路 411。于是，重组电路 411 根据节点中包含的 ID 编号 OD - ID 把相应 AV 数据作为纹理附加于所述节点。

- 25 然而，如果取出的 URL 指定网络中的另一个服务器存储的文件(URL 是具有例如表达式(1)格式的字符串)，则 URL 变化电路 412 把该信息提供给多路分解电路 404，然后多路分解电路 404 向该服务器发送文件传输请求。于是，多路复用比特流 FS' 通过类似的方法被传输，并进行显示。

- 根据上实施例，即使拟附加于节点的 AV 数据(比特流)ES 存在于网络的  
30 另一服务器中，也可以获得并显示期望的 AV 数据(比特流)。

下面参考附图 19，描述根据本发明第三实施例的编码装置。图 19 是表





示根据本发明第三实施例编码装置的方块图。在图 19 中，与图 1 中相对应的部分在后面采用相同的参考标记并不再描述。

图 19 的实施例中，在图 1 的实施例中加入 URL 变化电路 309、开关 310 和控制电路 311。而且，分析电路 307 输出的数据和 OD 生成电路 304 输出的 ID 编号 OD - ID 被提供给 URL 变化电路 309。URL 变化电路 309 输出的数据和 OD 生成电路 304 输出的 ID 编号 OD - ID 被提供给开关 310，而控制电路 311 控制开关 310。其余的结构与图 1 中的实施例相同。

URL 变化电路 309 把 OD 生成电路 304 输出的 ID 编号 OD - ID 变换为相应的 ASCII 码格式的字符串，并输出之。因为在图 15 的第二实施例中描述了 URL 变化电路 309 的工作情况，这里不再描述。

由控制电路 311 控制，开关 310 从 OD 生成电路 304 输出的 ID 编号 OD - ID 和 URL 变化电路 309 输出的 URL 中选择一个，并把所选的 OD - ID 或 URL 输出给 BIFS 编码器 308。控制电路 311 根据例如应用的类型控制开关 310 的转换。下面简要描述本实施例的工作情况。

其格式已经被 URL 变化电路 309 转换的 URL(细节在第二实施例中已经描述，在此不再描述)被提供给开关 310。类似地，由 OD 生成电路 304 输出的 ID 编号 OD - ID 被提供给开关 310。

在控制电路 311 的控制下来改变开关 310 的连接。例如对于实时通讯或硬件设计而言其优点在于，ID 编号 OD - ID 被直接描述为例如 10 位标记位形式的数字而不是字符串。因此，在这样的应用中，开关 310 由开关控制电路 311 控制以便从 OD 生成电路 304 选择输出数据，在这种情况下，由 BIFS 编码器 308 把 ID 编号 OD - ID 记录在二进制格式的场景描述 B - SD 中。如果将由 URL 指定的 AV 数据(比特流)存储在网络的另一服务器中，则控制电路 311 控制开关 310，改变它的连接，以便 URL 变化电路 309 的输出数据得以被选择，从而 URL 输出到 BIFS 编码器 308 并被记录。

另一方面，在计算机应用的情况下，其优点在于，流由 URL 字符串指定，这是因为高度的灵活性。因此，在这样的应用中，控制开关 310，使之连接到 URL 变化电路 309 上，从而由 BIFS 编码器 308 把 URL 记录在二进制格式的场景描述 B - SD 中。

图 20 表示作为纹理附加用的移动图像的场景描述 B - SD 二进制格式的实例。在图 20 中，第 29 行和第 30 行的“isString”是表示描述的是 ID

编号 OD - ID 还是 URL 的 1 位标记位。如果该值为“0”，则 10 位的 ID 编号 OD - ID 被记录在节点中。如果“isString”的值为“1”，记录的是 URL。该 URL 是已经被 URL 变化电路 309 改写的字符串，以便表示拟附加于节点的移动图像的 ID 编号 OD - ID。

- 5 图 21 表示作为纹理附加用的静止图像的场景描述 B - SD 二进制格式的实例。在该图中，如同上面的情况一样，在第 17 行和第 18 行上的“isString”是表示所描述的是 ID 编号 OD - ID 还是 URL 的 1 位标记位。

已经被上述编码装置编码的多路传输流 FS 经过传输线传输到解码装置。

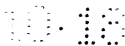
- 10 图 22 是表示与根据本发明的图 19 的编码装置相应的解码装置的第三实施例的方块图。在图 22 中，与图 8 中相应的部分在后面采用相同的参考标记并不再描述。

图 22 的实施例，在图 8 实施例中加入 URL 变化电路 412。其余的结构与图 8 中的实施例相同。

- 15 在本实施例中，分析电路 410 解码“isString”。如果该值为“1”，则分析电路 410 向 URL 变化电路 412 提供一个 URL。如果该值为“0”，则分析电路 410 解码 ID 编号 OD - ID 并向重组电路 411 提供一个结果。

- 如果所述 URL 是以例如表达式(2)的形式描述的，则 URL 变化电路 412 解码 ID 编号 OD - ID，并向重组电路 411 输出一个结果。如果所述 URL 表  
20 示存在于另一服务器中的文件，则所述信息提供给多路分解电路 404，而且该多路分解电路 404 访问该服务器并读出需要的文件。

- 下面简要描述本实施例的工作情况。读出的场景描述 SD(节点)被提供给分析电路 410 并在此进行分析。将被分析的场景描述提供给重组电路 411。另外，分析电路 410 解码“isString”并判断它的值是否为“1”。如果判断  
25 该值为“1”，其分析电路 410 把将要作为纹理被附加于节点的 AV 数据(比特流)的 URL 提供给 URL 变化电路 412。如果所述 URL 是以例如表达式(2)的形式描述的(例如字符串的头部为“mpeg4”)，则 URL 变化电路 412 解码 ID 编号 OD - ID，并输出给重组电路 411，所述 ID 编号 OD - ID 是来自字符串的目标描述 OD 的 ID。如果所述 URL 指定存在于另一服务器中的文  
30 件，则所述信息被提供给多路分解电路 404，并且该多路分解电路 404 访问该服务器，要求该服务器传输需要的文件，并接收该文件。即使当使用多个



服务器通讯时，每个服务器都按照上述方式工作。

另一方面，如果“isString”为“0”，则分析电路410解码ID编号OD - ID并向重组电路411输出一个结果。其余的工作情况与第一实施例相同，后面不再描述。

5 根据上述实施例，可以根据应用的类型选择最适当的编码方法。

根据本发明，例如磁盘、DVD - R、CD - R、CD - ROM等记录介质含有上述编码方法产生的编码图像信号；当从记录介质上再现图像时解码这些编码图像信号。

10 虽然已经参考方块图示出了本发明的编码和解码装置及方法，而且对每一个方块图提供了不同的物理元件，因此可以在为多用途而编程的多用途(通用)计算机上实现这种方法和装置。在这方面，所述记录介质或其他存储装置都能包含完成上述编码和解码操作方法中提出的每个步骤的操作指令(源程序码或软件)，还应该指出的是，与通讯网或类似线路(例如互联网、数字卫星等等)相连的传输通道可以用于接收和传输来自编码器的数据，并解码这种  
15 编码的数据。

举例来说，根据本发明的编码和解码装置方法可被用于编码和解码来自数字视盘、图像数据库、图像压缩和展开单元、从互联网卸载的图像或者实现这些系统的软件模块的信息。

在编码装置中，输入编码方法和记录介质、三维空间的模型数据(VRML  
20 数据)，是输入数据(AV数据流)。提取输入的三维空间模型数据(VRML数据)的节点中所包括的位置标识数据(URL)。将提取的位置标识数据(URL)转换为与位置标识数据(URL)指定的数据(AV数据流)相应的流ID。用通过转换获得的流ID替代节点的位置标识数据(URL)。由替代所得的三维空间模型数据(VRML数据)和AV数据被多路复用到同一流中。因此，能够传输作为三维  
25 空间模型数据(VRML数据)而被描述的目标和根据例如MPEG方案压缩的自然图像；将它们多路复用到同一流中的状态进行这种传输。

在解码装置中，从多路复用数据中取出解码方法和记录介质以及节点，并从多路复用数据中取出数据(AV数据流)。从各节点取出表示节点与数据(AV数据流)之间相关特性的信息。根据取出的表示相关特性的信息，排序(匹  
30 配)各节点与数据(AV数据流)。根据相关特性结果合成所述节点和数据(AV数据流)。因此，能够解码以如下状态传输的数据，即使得描述为三维空间模型

数据(VRML 数据)的目标和根据例如 MPEG 方案压缩的自然图像被多路复用到同一流中。

- 因此应该看到,上面提到的目的通过前面的描述变得更清楚,并可以充分地实现,而且因为在实现上述方法以及提出的结构中可以进行某些变化,而不致离开本发明的实质和范围,所以上面描述中包含的和附图中示出的所有情况都是解释性说明,实际并不限于这些。
- 5

还应该理解后面各权利要求涵盖这里描述的本发明的所有一般和具体特征;用术语来说,可以说本发明范围的所有叙述,均可落入权利要求书中,

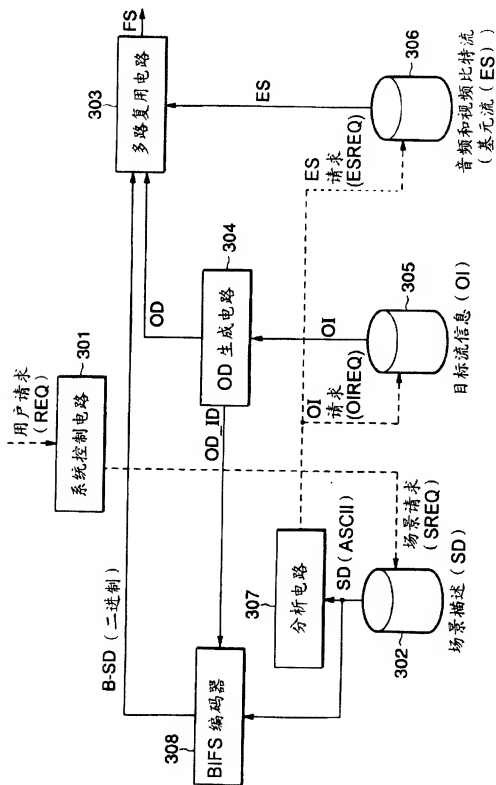


图 1

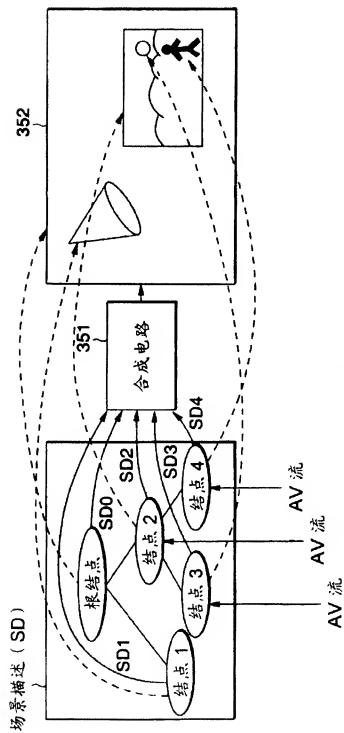


图 2

```
移动纹理 {  
  exposedField SFBool loop FALSE  
  exposedField SFFloat speed 1  
  exposedField SFTime startTime 0  
  exposedField SFTime stopTime 0  
  exposedField SFString URL "http://serverA/AV_scene.1/object_file.1"  
  field SFBool repeatS TRUE  
  field SFBool repeatT TRUE  
  eventOut SFFloat duration_changed  
  eventOut SFBool is Active  
}
```

图 3

```
图象纹理 {  
  exposedField SFString URL "http://serverA/AV_scene.1/object_file.1"  
  field SFBool repeatS TRUE  
  field SFBool repeatT TRUE  
}
```

图 4

## 节点语句

```

class MovieTexture extends Node : bit(7) nodeType=0100011{
    bit(1) is Reused ; // This bit describes whether this node is a
    // reused node or a newly defined one. This is equivalent to USE
    if(isReused) {
        bit(10) nodeID ; // The NodeID to be re-used
    }else {
        bit(7) nodeType ;
        bit(1) is Updatable
        if(isUpdatable){
            bit(10) nodeID ;
        }
        bit(1) endFlag ; // Are there any arguments at all ?
        while (!endFlag){

            bit(3)      fieldRef ; // index to the field to be specified below
            switch(fieldRef){
            case 0 :
                SFBool loop ;
                break ;

            case 1 :
                SFFloat speed ;
                break ;

            case 2 :
                SFTime startTime ;
                break ;

            case 3 :
                SFTime stopTime ;
                break ;

            case 4 :
                SFObjectID objectDescriptorID ;
                break ;

            case 5 :
                SFBool repeatS ;
                break ;

            case 6 :
                SFBool repeatT ;
                break ;
            }
            bit(1) endFlag ; // Are there more arguments ?
        }
    }
}

```

图 5



## 节点语句

```

class ImageTexture extends Node : bit(7) nodeType=0011100{
    bit(1) is Reused ; // This bit describes whether this node is a
    // reused node or a newly defined one. This is equivalent to USE
    if(isReused) {
        bit(10) nodeID ; // The NodeID to be re-used
    }else {
        bit(7) nodeType ;
        bit(1) is Updatable
        if(isUpdatable){
            bit(10) nodeID ;
        }
        bit(1) endFlag ; // Are there any arguments at all ?
        while (!endFlag){

            bit(2)      fieldRef ; // index to the field to be specified below
            switch(fieldRef){
            case 0 :
                SFOBJECTID objectDescriptorID ;
                break ;

            case 1 :
                SFBool repeatS ;
                break ;

            case 2 :
                SFBool repeatT ;
                break ;
            }
            bit(1) endFlag ; // Are there more arguments ?
        }
    }
}

```

图 6

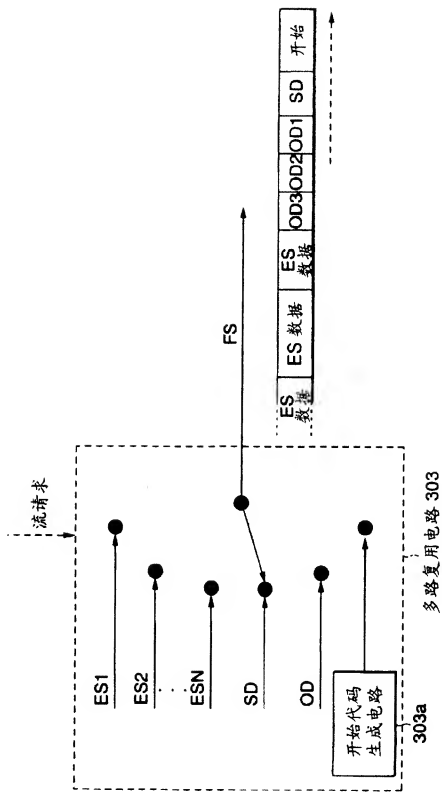


图 7

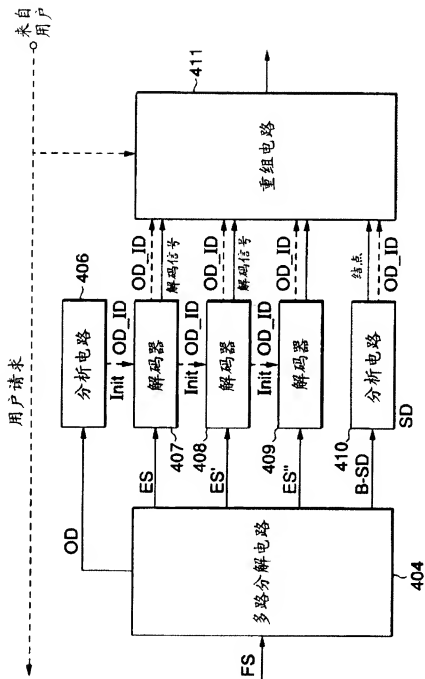
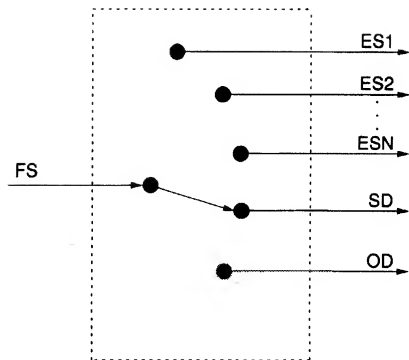


图 8



多路分解电路 404

图 9

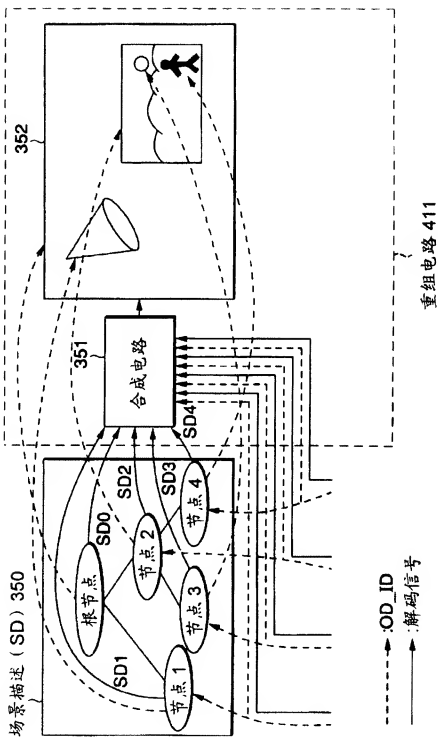


图 10

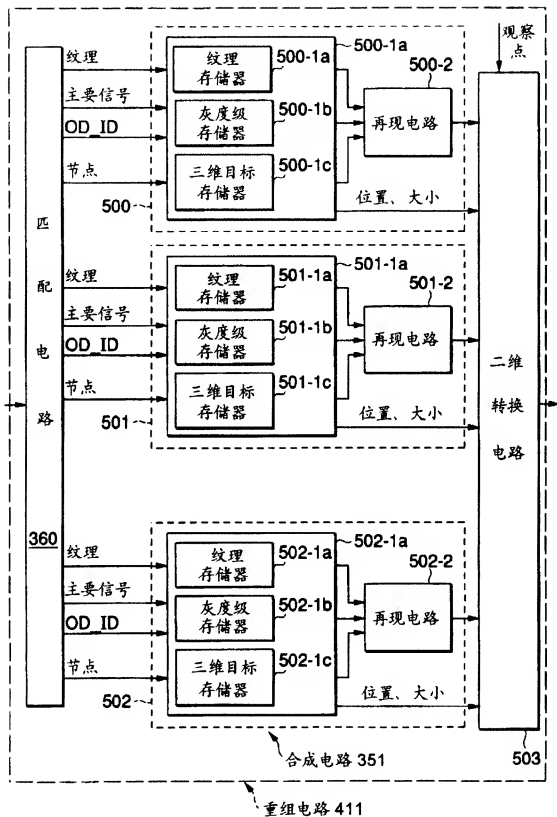


图 11

景物描述符 ()	
{	
NodeId	10
streamCount	5
extensionFlag	1
for(j=0 ; j<streamCount ; j++)	
{	
ES_Descriptor()	
}	
if(extensionFlag==1)	
{	
descriptorCount	8
for(j=0 ; j<descriptorCount; j++)	
{	
Descriptor()	8
}	
}	
}	

表 0-1: 目标描述符语句

图 12

ES_Descriptor()	
{	
ES_Number	5
reserved	2
extensionFlag	1
streamType()	8
QoS_Descriptor()	8
ESConfig Params()	8
if(extensionFlag==1)	
{	
descriptorCount	
for(j=0 ; j<descriptorCount ; j++)	
{	
Descriptor()	8
}	
}	
}	

表 0-2: ES\_Descriptor 语句

图 13



ESConfigParams()	
{	
predefined	8
if(predefined==0)	
{	
bufferSizeEB	24
useTimeStamps	1
useOCR	1
timeStampResolution	32
OCRResolution	32
timeStampLength	6
OCRLength	6
if(!useTimeStamps)	
{	
accessUnitDecodingRate	16
accessUnitPresentationRate	16
startDecodingTimeStamp	timeStampLength
startPresentationTimeStamp	timeStampLength
}	
AU_Length	4
instantBitrateLength	8
streamPriority	5
degradationPriorityLength	4
seqNumLength	4
}	
}	

表 0-3: ES\_ConfigParams 语句

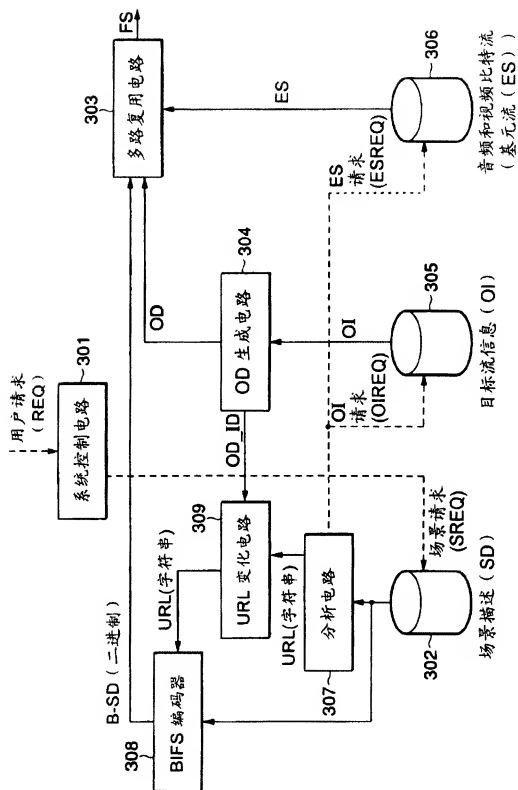


图 15

## 节点语法

```

class MovieTexture extends Node : bit(7) nodeType=0100011{
    bit(1) is Reused ; // This bit describes whether this node is a
    // reused node or a newly defined one. This is equivalent to USE
    if(isReused) {
        bit(10) nodeID ; // The NodeID to be re-used
    } else {
        bit(7) nodeType ;
        bit(1) is Updatable
        if(isUpdatable){
            bit(10) nodeID ;
        }
        bit(1) endFlag ; // Are there any arguments at all ?
        while (!endFlag){

            bit(3)    fieldRef ; // index to the field to be specified below
            switch(fieldRef){
            case 0 :
                SFBool loop ;
                break ;

            case 1 :
                SFFloat speed ;
                break ;

            case 2 :
                SFTime startTime ;
                break ;

            case 3 :
                SFTime stopTime ;
                break ;

            case 4 :
                SFString URL ;
                break ;

            case 5 :
                SFBool repeatS ;
                break ;

            case 6 :
                SFBool repeatT ;
                break ;
            }
            bit(1) endFlag ; // Are there more arguments ?
        }
    }
}

```

图 16

图象纹理

节点语法

```

class ImageTexture extends Node : bit(7) nodeType=0011100{
    bit(1) is Reused ; // This bit describes whether this node is a
    // reused node or a newly defined one. This is equivalent to USE
    if(isReused) {
        bit(10) nodeID ; // The NodeID to be re-used
    }else {
        bit(7) nodeType ;
        bit(1) is Updatable
        if(isUpdatable){
            bit(10) nodeID ;
        }
        bit(1) endFlag ; // Are there any arguments at all ?
        while (!endFlag){

            bit(2)      fieldRef ; // index to the field to be specified below
            switch(fieldRef){
            case 0 :
                SFString URL ;
                break ;

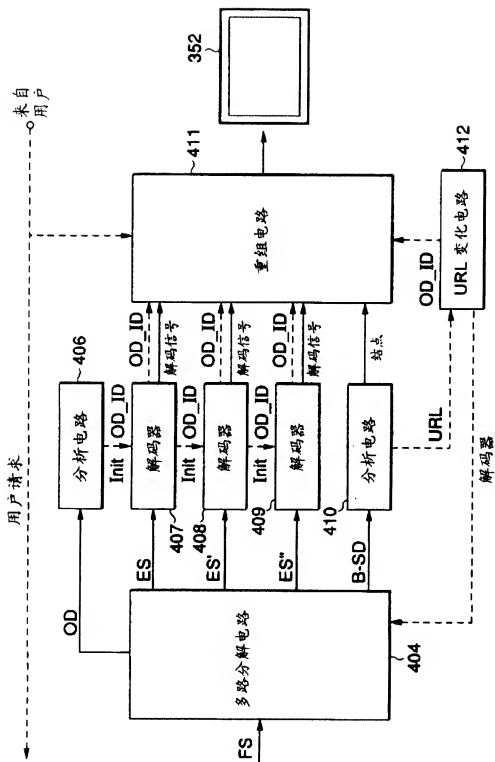
            case 1 :
                SFBool repeatS ;
                break ;

            case 2 :
                SFBool repeatT ;
                break ;

            }
            bit(1) endFlag ; // Are there more arguments ?
        }
    }
}

```

图 17



18

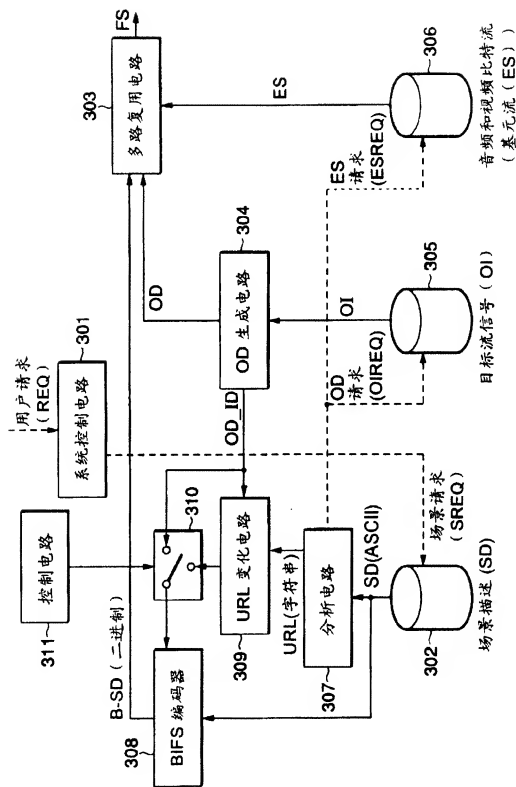


图 19

# 节点语句

```

class MovieTexture extends Node : bit(7) nodeType=0100011{
    bit(1) is Reused ; // This bit describes whether this node is a
    // reused node or a newly defined one. This is equivalent to USE
    if(isReused) {
        bit(10) nodeID ; // The NodeID to be re-used
    }else {
        bit(7) nodeType ;
        bit(1) is Updatable
        if(isUpdatable){
            bit(10) nodeID ;
        }
        bit(1) endFlag ; // Are there any arguments at all ?
        while (!endFlag){
            bit(3)      fieldRef ; // index to the field to be specified below
            switch(fieldRef){
                case 0 :
                    SFBool loop ;
                    break ;
                case 1 :
                    SFFloat speed ;
                    break ;
                case 2 :
                    SFTime startTime ;
                    break ;
                case 3 :
                    SFTime stopTime ;
                    break ;
                case 4 :
                    bit(1) isString
                    if(isString)
                        SFString URL ;
                    else
                        SFObjectID ObjectDescriptorID ;
                    break ;
                case 5 :
                    SFBool repeatS ;
                    break ;
                case 6 :
                    SFBool repeatT ;
                    break ;
            }
            bit(1) endFlag ; // Are there more arguments ?
        }
    }
}

```

图 20

## 节点语句

```

class ImageTexture extends Node : bit(7) nodeType=0011100{
    bit(1) is Reused ; // This bit describes whether this node is a
    // reused node or a newly defined one. This is equivalent to USE
    if(isReused) {
        bit(10) nodeID ; // The NodeID to be re-used
    } else {
        bit(7) nodeType ;
        bit(1) is Updatable
        if(isUpdatable){
            bit(10) nodeID ;
        }
        bit(1) endFlag ; // Are there any arguments at all ?
        while (lendFlag){

            bit(2) fieldRef ; // index to the field to be specified below
            switch(fieldRef){
                case 0 :
                    bit(1) isString
                    if(isString)
                        SFString URL ;
                    else
                        SFObjectID ObjectDescriptorID ;
                    break ;
                case 1 :
                    SFBool repeatS ;
                    break ;
                case 2 :
                    SFBool repeatT ;
                    break ;
            }
            bit(1) endFlag ; // Are there more arguments ?
        }
    }
}

```

图 21



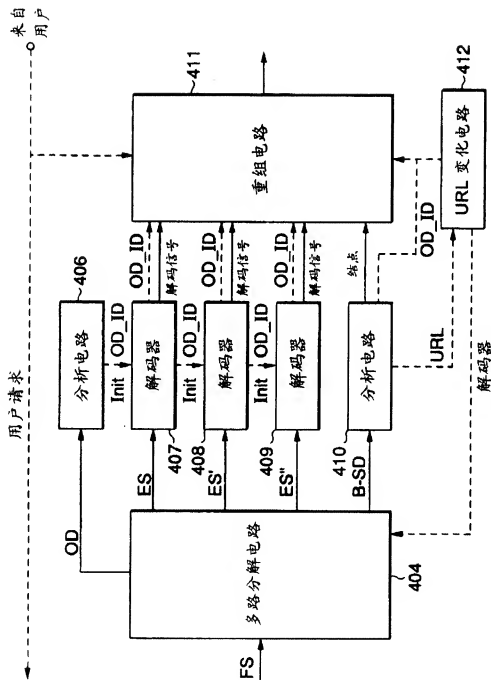


图 22

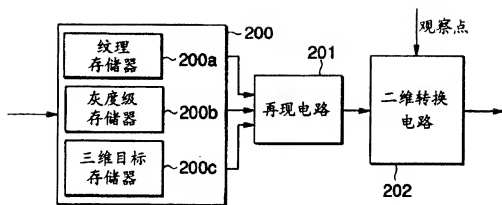


图 23

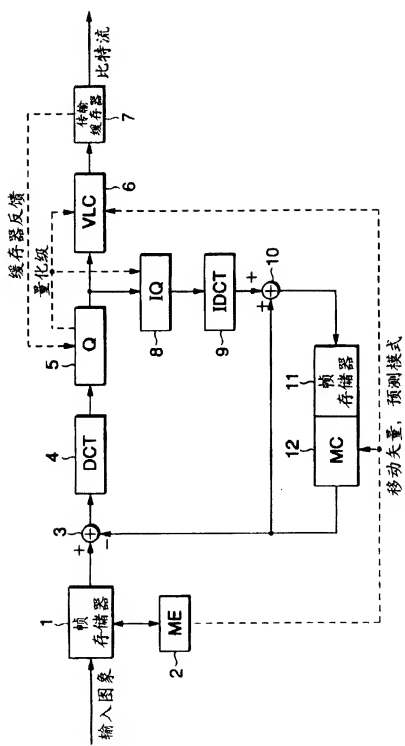


图 24 图象信号编码装置

图 24

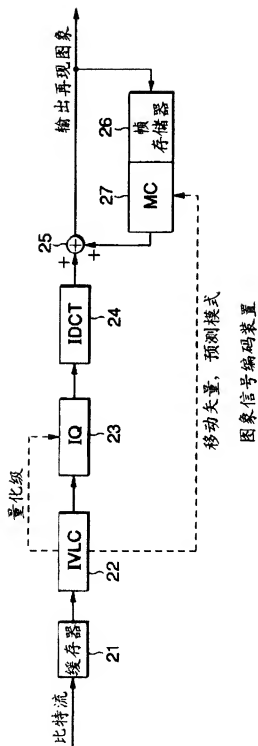


图 25

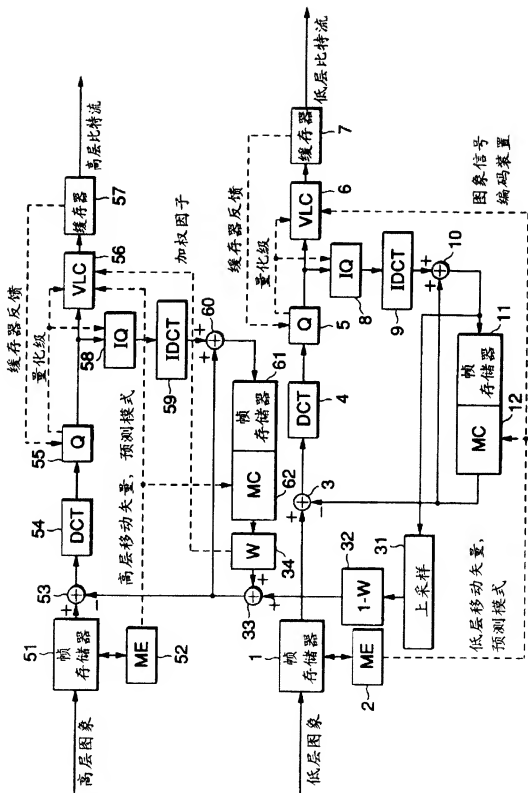


图 26

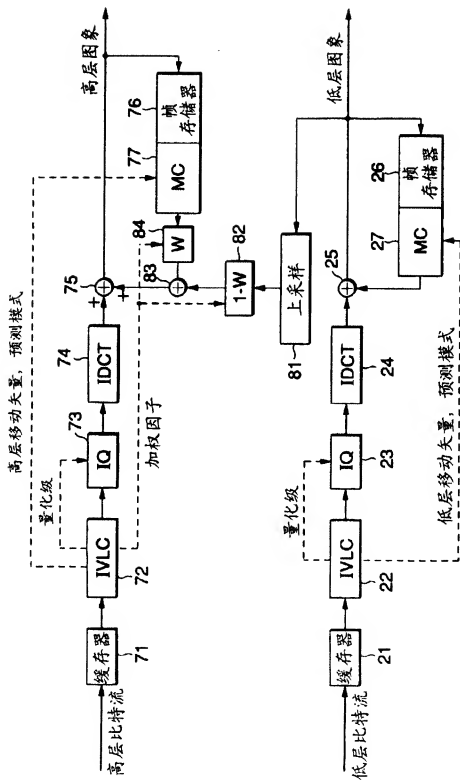


图 27

图 27